CS 405 Project 2: Textures + Illumination – Alara Basak 23546

Introduction

This report describes the enhancements made to the CS 405 Project 2 involving textures and illumination. The project's primary objectives were to modify the setTexture method to handle non-power-of-2 sized textures and to implement basic lighting in the scene.

Task 1: Texture Modification

The setTexture function was updated to handle textures with width and height that are not a power of two. This involved changing the texture parameters for wrapping and minifying filter when the texture is non-power-of-2. The updated function now checks if the image dimensions are a power of two. Else (which is implemented by me), it uses CLAMP_TO_EDGE for wrapping and sets the minifying filter to LINEAR. You can see the updated codes below:

```
setTexture(img) {
        ...
    // Set texture parameters
    if (isPowerOf2(img.width) && isPowerOf2(img.height)) {
        // Only generate mipmaps for power-of-2 textures
        gl.generateMipmap(gl.TEXTURE_2D);
    } else {
        // implemented logic for non power of two textures:
        // CLAMP_TO_EDGE
        gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_WRAP_S, gl.CLAMP_TO_EDGE);
        gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_WRAP_T, gl.CLAMP_TO_EDGE);
        gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MIN_FILTER, gl.LINEAR);
    }
        ...
}
```

Task 2: Basic Lighting Implementation

The project's second task involved implementing basic lighting for the scene. This required modifications to the constructor, setMesh, draw, setAmbientLight, enableLighting functions, and the fragment shader. The lighting model includes ambient and diffuse lighting. The light's position can be changed using arrow keys, and the ambient light intensity is adjustable via a slider in the UI. Modified codes can be seen in github/document.

Methodology

The implementation started with updating the MeshDrawer class to support new lighting features. In the constructor, uniform locations for lighting parameters like light position and ambient light were initialized. The setMesh method was modified to handle normal coordinates for lighting calculations. The draw method was updated to calculate and pass the lighting parameters to the shader. The fragment shader (meshFS) was significantly altered to incorporate lighting calculations, including ambient and diffuse components.

Results and Conclusion

The modifications to the project2.js file resulted in a successful implementation of the required features. The application now supports textures of non-power-of-2 dimensions and displays basic lighting effects on the 3D models. The interactive control over light position and ambient light intensity demonstrates the dynamic nature of the lighting in the scene.