Implementation of Texture Handling and Lighting in WebGL

Tuna Karagül-28027

Introduction

This report outlines the methodologies used in enhancing a WebGL-based graphics application. The focus was on two primary objectives: enabling the application to handle textures with non-power-of-two dimensions and incorporating basic ambient and diffuse lighting to improve the visual quality of the scene.

Texture Handling

Problem Statement:

The initial implementation restricted texture inputs to images whose dimensions were powers of two, limiting the flexibility in texture usage.

Solution Approach:

To address this, the texture parameter settings in the WebGL context were modified. This involved adjusting texture wrapping and filtering options to support textures with non-power-of-two dimensions.

Implementation:

The setTexture method was updated to check the dimensions of the texture. For textures not adhering to the power-of-two requirement, gl.CLAMP_TO_EDGE was used for both gl.TEXTURE_WRAP_S and gl.TEXTURE_WRAP_T. Additionally, gl.LINEAR was set for gl.TEXTURE_MIN_FILTER to ensure proper texture rendering.

Lighting

Problem Statement:

The application initially lacked any form of lighting, resulting in a flat and unrealistic appearance of the scene.

Solution Approach:

The introduction of basic ambient and diffuse lighting was deemed necessary to simulate more realistic lighting effects.

Implementation:

Vertex Shader: Adjustments were made to pass the normal vectors to the fragment shader.

Fragment Shader: Significant changes were implemented to calculate ambient and diffuse lighting. This was based on the light's position, the normal vectors of the surface, and the ambient light intensity defined.

JavaScript Integration: Uniform variables for light position, ambient intensity, and other lighting properties were added to the JavaScript code. These were dynamically updated in the rendering loop.

Results

The implementation successfully allowed the application to support more flexible texture inputs. Moreover, the introduction of basic lighting effects significantly enhanced the visual appeal of the scene.

Conclusion

The enhancements made to the WebGL application effectively addressed the critical areas of

texture handling and lighting. This resulted in a more versatile and visually appealing graphical application, demonstrating a significant improvement over the initial version.