CS 405 Project 2 Report

Mert Ali Çelik 26958

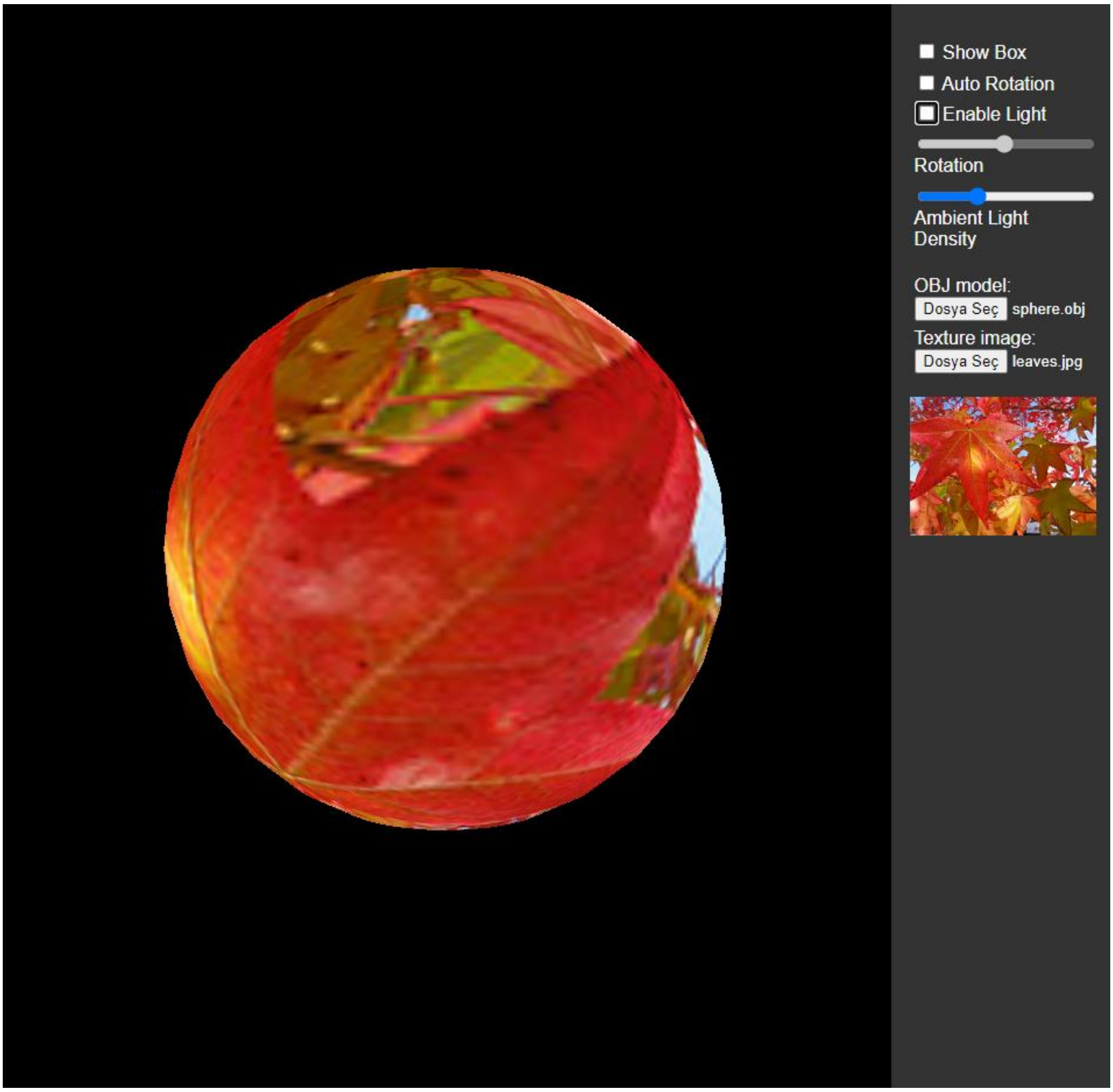
Introduction

In this project, I worked on two tasks to enhance a WebGL application. In Task 1, I adjusted the setTexture function to handle images with dimensions as powers of two, ensuring compatibility. Moving on to Task 2, I introduced basic lighting effects like ambient and diffuse lighting to bring realism and depth to the 3D scenes. Let's delve into the details of each task and the improvements made to create a more engaging WebGL experience.

Task 1

In task 1, i changed the setTexture inside the project2.js to accept pictures which have width and height values of a power of two. I used parameters such as g1.TEXTURE_TO_EDGE to be able to use it as a power of two. And also g1.generateMipMap was not supported by power of two textures so i used a compatible one which is g1.LINEAR.

After that, when i try it on my html page with the given leaves.png test i see these outputs.



☐ Show Box

☐ Auto Rotation

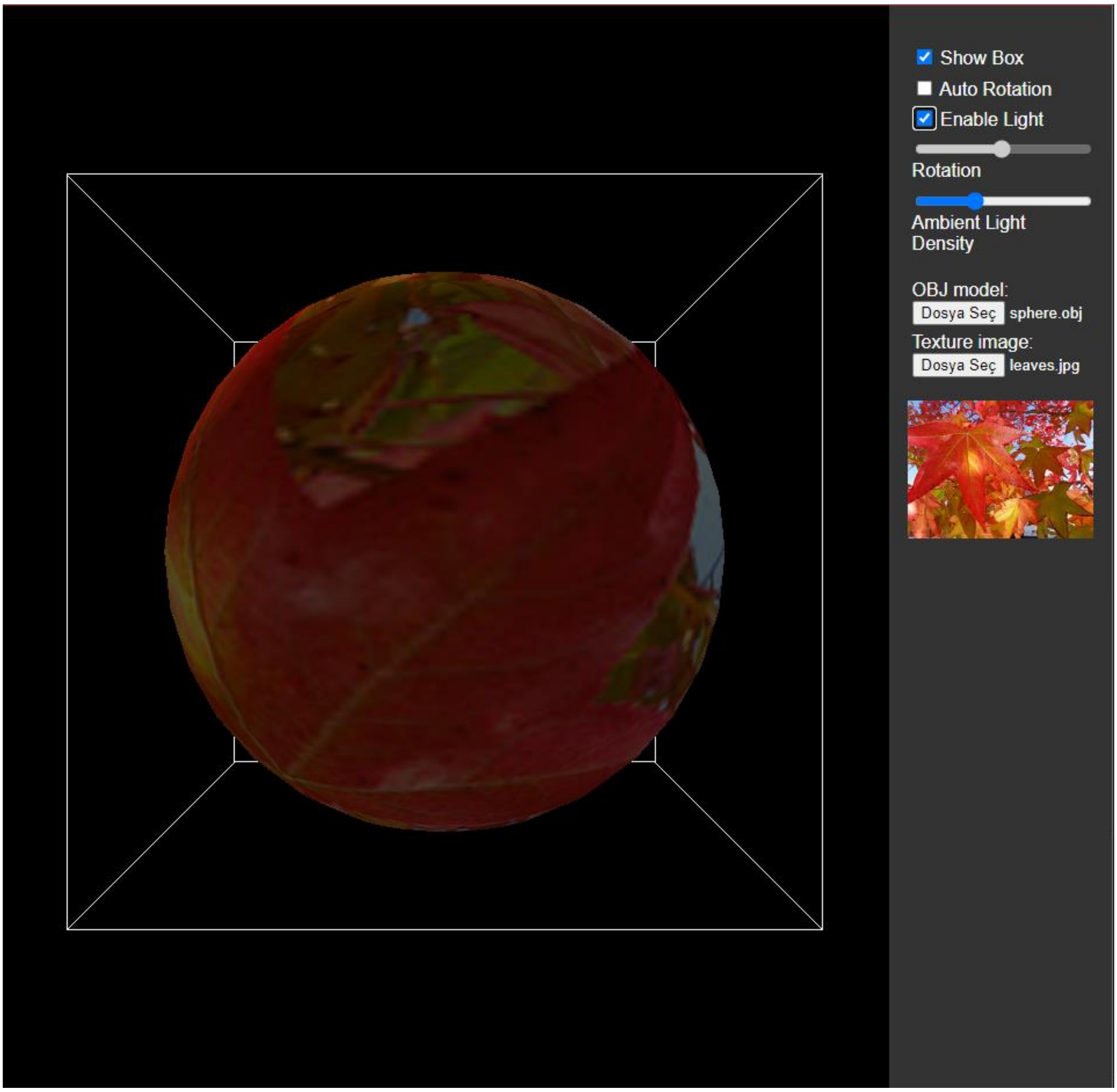
☐ Enable Light

Rotation

Ambient Light Density

OBJ model:
Dosya Seç sphere.obj

Texture image:
Dosya Seç leaves.jpg



☑ Show Box

□ Auto Rotation

☑ Enable Light

Rotation

Ambient Light Density

OBJ model: Dosya Seç sphere.obj

Texture image: Dosya Seç leaves.jpg

Task 2

While working on Task 2, i made several changes to include basic lighting effects like ambient and diffuse lighting. These changes are essential for adding a more realistic and layered feel to the 3D scenes in WebGL.

The setMesh method was updated to accommodate normal vectors, a key component for lighting computations. Within the MeshDrawer class, expansion occurred to incorporate essential WebGL uniform and attribute locations related to lighting. This encompassed obtaining uniform locations for light position (lightPosLoc), ambient light intensity (ambientLoc), a flag for enabling or disabling lighting (enableLightingLoc), and an attribute location for vertex normals (normalLoc). These additions laid the groundwork for conducting lighting calculations in the shader programs.

Additionally, two pivotal functions, enableLighting and setAmbientLight, were implemented to dynamically control the lighting. The enableLighting function allows toggling the lighting effect on and off, while setAmbientLight adjusts the intensity of ambient light based on user input from an HTML range slider. An essential correction was made in the setAmbientLight method, ensuring the correct parameter usage to accurately influence the scene's ambient light intensity by the slider's value.

Within the draw method, the application now establishes the light position and dynamically updates it based on user interactions, facilitating dynamic lighting effects. The most substantial change on the shader side occurred in the fragment shader (meshFS), where the code was adapted to calculate both diffuse and ambient lighting. Diffuse lighting imparts a sense of directionality and shadow, while ambient lighting establishes a foundational light level irrespective of light direction.

These advancements synergistically elevate the WebGL application, delivering a heightened sense of dynamism and visual allure. They effectively showcase foundational aspects of lighting in 3D graphics, enhancing the overall immersive experience for users.