General Modifications Overview

Task 1: SceneNode Draw Function

Implemented the draw function in the SceneNode class to handle hierarchical transformations and recursive rendering. This ensured each node applied its own transformations and passed them to its children.

Task 2: Diffuse and Specular Lighting

Enhanced the fragment shader in meshDrawer.js by adding calculations for diffuse and specular lighting. These updates created realistic light effects based on the Phong lighting model.

Task 3: Adding Mars

Added Mars to the solar system by creating a SceneNode in .html with a sphere mesh and texture. Applied specific transformations for position, scale, and rotation and linked Mars as a child of the Sun node.

Task 1 Methodology: SceneNode Draw Function

The purpose of Task 1 is to implement the draw function to handle hierarchical rendering in the scene graph. Each node applies its own transformations, combines them with those of its parent, and passes the result to its children. This ensures a proper propagation of transformations throughout the graph

1. **Apply Transformations**
   Each node takes its own transformations (like position, rotation, and size) and combines them with the parent’s transformations.
2. **Render the Node**
   If the node has a mesh to draw, it is rendered using the combined transformations.
3. **Draw Children**
   The updated transformations are passed to all child nodes, which repeat the process.

This ensures all nodes and their children are positioned and drawn correctly in the scene.

Task 2 Methodology: Diffuse and Specular Lighting

1. **Understanding the Goal**: Added diffuse and specular lighting to the fragment shader for realistic light effects.
2. **Diffuse Lighting**: Calculated the dot product of the normalized surface normal and light direction to simulate light scattering.
3. **Specular Lighting**: Used the reflection of the light direction and the view direction, raised to a Phong exponent, for shiny highlights.
4. **Combining Components**: Combined ambient, diffuse, and specular lighting for the final fragment color.
5. **Testing**: Verified correct rendering of diffuse and specular effects with varying Phong exponent values.

This approach ensured enhanced lighting while adhering to shader constraints.

Task 3 Methodology

1. **Creating Mars**:
   We initialized a MeshDrawer for Mars to handle its rendering. The sphere mesh data (position, texture coordinates, normals) was assigned to the MeshDrawer, and the provided texture link was used to set the Mars surface.
2. **Configuring Transformations**:
   Using a TRS object:
   - Mars was translated by -6 units along the X-axis to position it relative to the Sun.
   - It was scaled uniformly to 0.35 to reflect its size in the scene.
   - Mars's Z-axis rotation was set to 1.5 * zRotation, creating a dynamic rotation effect.
3. **Adding to the Scene**:
   A SceneNode was created for Mars, linking it as a child to the Sun node. This ensures Mars inherits the Sun’s transformations, maintaining its orbit relative to the Sun.
4. **Testing**:
   We tested Mars in the solar system, confirming its texture, position, scale, and rotation behaved as intended, completing its integration into the scene.