Project 3 Report

Task 1 of the project required the implementation of the `draw` function for the `SceneNode` class, which is crucial for rendering objects within the WebGL context. My code successfully constructs the transformation matrices for a node and recursively for its children, applying model, view, and projection transformations. This allows for the hierarchical modeling necessary to represent complex scenes, such as a solar system, where objects maintain spatial relationships and transformations relative to their parent nodes. The code fulfills the task's requirements by ensuring that transformations are correctly propagated through the scene graph, resulting in a correctly rendered 3D scene.

In Task 2 of the project, the objective was to enhance the fragment shader within the **meshDrawer.js** class to include diffuse and specular lighting, thereby providing a more realistic rendering of objects with considerations for light interaction. The diffuse lighting was computed based on the Lambertian reflectance model, which determines the color intensity at a point on a surface relative to its alignment with the light source. The specular component was added to simulate the bright, reflective highlights that occur when light is reflected in a concentrated manner, typically on glossy surfaces. The Phong reflection model was employed for this purpose, taking into account the viewer's position and the shininess of the surface. The completion of this task involved writing GLSL code to calculate these lighting effects and integrate them into the rendering pipeline. The resulting visual enhancement significantly improved the realism of the scene, particularly in how light interacts with the surfaces of objects within the solar system simulation.

Task 3 of the project involved integrating a new celestial body, Mars, into the existing solar system simulation. The task was accomplished by extending the scene graph with a new node representing Mars. The implementation required creating a new instance of **MeshDrawer** and setting up Mars with appropriate mesh data, texture, and transformation matrices for translation, scaling, and rotation. The simulation now accurately depicts Mars in orbit, with a scaling factor of 0.35 and a translation of -6 units on the X-axis relative to the Sun. Mars rotates on its axis at a speed of 1.5 times the base rotation, enhancing the simulation's dynamism and realism.

For Task 3 of the project, the goal was to simulate Mars within the virtual solar system. The process involved updating the existing scene graph by adding Mars as a new node. This required careful adjustments to the WebGL code to ensure Mars was correctly placed in relation to the Sun and had the appropriate texture, scale, and rotational behavior to reflect its real-world counterpart. The resulting simulation offers a more comprehensive representation of the solar system, enhancing the educational value of the project by providing a more complete celestial model. The successful integration of Mars demonstrates the extendibility of the scene graph structure and the power of WebGL for creating interactive 3D simulations.