Task 1

The implementation was straightforward thanks to the provided `MatrixMult` and `getTransformationMatrix` functions in the utilities. A single matrix multiplication with the `transformMatrix` was sufficient to handle all the parameters. Additionally, the `draw` function needed to recursively render all child elements, while passing the parent's parameters down the hierarchy.

Task 2

Diffuse lighting depends on the direction of the light and the surface normal, and it can be implemented using a simple dot product. To ensure the result is non-negative, the maximum of the dot product and zero is taken. Specular lighting can be calculated in either world space or view space; however, view space is preferred in this case since we cannot modify other parts of the fragment shader and the camera's position/direction is not provided in the file. The specular lighting implementation follows a standard approach, with the power value set to 16.0 to make it noticeable without overpowering the diffuse light.

Task 3

Adding Mars to the scene was straightforward since the process was identical to that used for the other planets. A new `MeshDrawer` object was created for Mars, with its mesh set using the existing `sphereBuffers` and its texture assigned using the provided image link. A translation of -6 units along the x-axis and a uniform scaling of 0.35 were applied. Finally, Mars was added to the scene by creating a new `SceneNode` with its parent set to `SunNode`.