CS 405 Project 3: Scene Graph + Illumination

Yigit Kaan Tonkaz – 29154

This project aimed to create a WebGL-based solar system visualization using a scene graph structure. The tasks were divided into three parts:

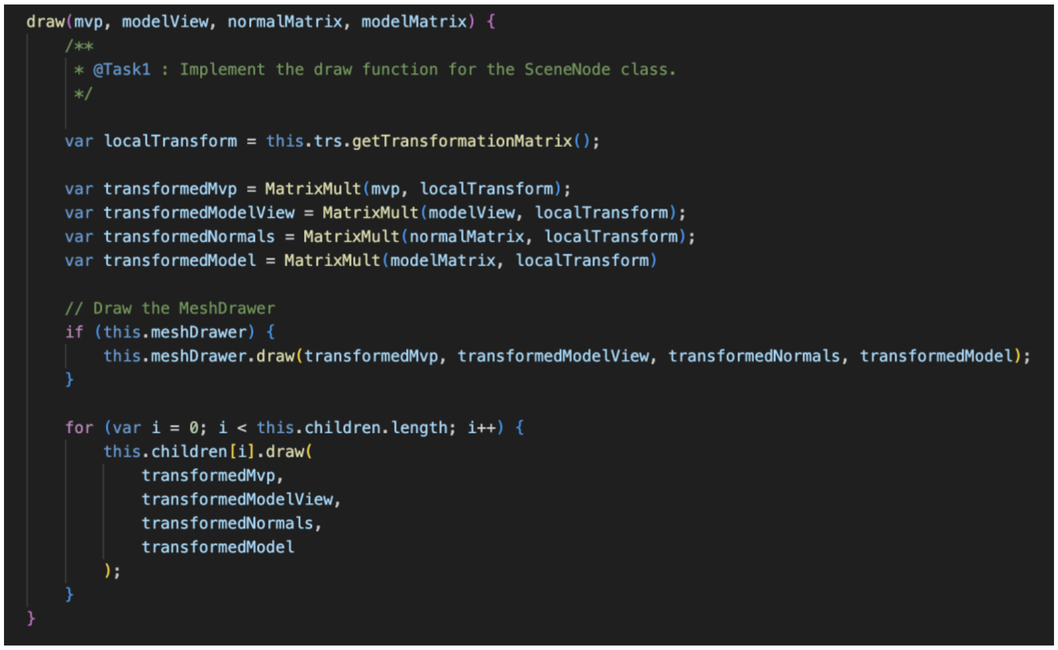
1. Implementing the draw method in SceneNode to traverse and render the scene graph.
2. Enhancing the MeshDrawer fragment shader to include diffuse and specular lighting.
3. Adding Mars as a new node to the scene graph and ensuring its transformations and textures were correct.

Task 1: Implementing the draw Method

The goal was to implement the draw function in the SceneNode class to recursively render all nodes in the scene graph while applying the correct transformations.

Implementation:

- Each node's local transformations (translation, rotation, scale) were applied using the TRS helper class. The getTransformationMatrix() method was used to compute the combined transformation matrix.
- This matrix was then multiplied with the parent transformations (model-view-projection, model-view, and normal matrices).
- After rendering the current node's mesh using the MeshDrawer class, the function recursively called draw on all child nodes, passing down the updated transformation matrices.



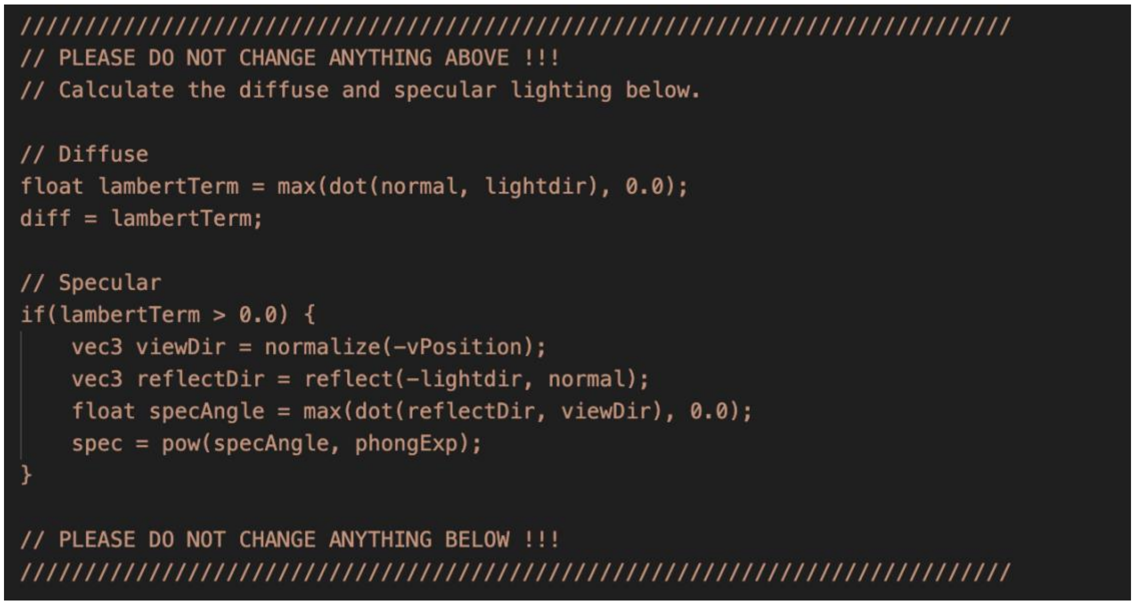
```
draw(mvp, modelView, normalMatrix, modelMatrix) {
    /**
     * @Task1 : Implement the draw function for the SceneNode class.
     */

    var localTransform = this.trs.getTransformationMatrix();

    var transformedMvp = MatrixMult(mvp, localTransform);
    var transformedModelView = MatrixMult(modelView, localTransform);
    var transformedNormals = MatrixMult(normalMatrix, localTransform);
    var transformedModel = MatrixMult(modelMatrix, localTransform)

    // Draw the MeshDrawer
    if (this.meshDrawer) {
        this.meshDrawer.draw(transformedMvp, transformedModelView, transformedNormals, transformedModel);
    }

    for (var i = 0; i < this.children.length; i++) {
        this.children[i].draw(
            transformedMvp,
            transformedModelView,
            transformedNormals,
            transformedModel
        );
    }
}
```

Task 2: Enhancing the Fragment Shader

The objective was to extend the fragment shader to calculate diffuse and specular lighting, providing a realistic lighting effect to the scene.

Implementation:

- Calculated using the Lambertian reflection model. This term represents how much light hits the surface directly, based on the angle between the light direction and the surface normal.
- Used the Phong reflection model. The reflect function was used to calculate the reflection vector, simulating shiny highlights where light reflects toward the viewer.



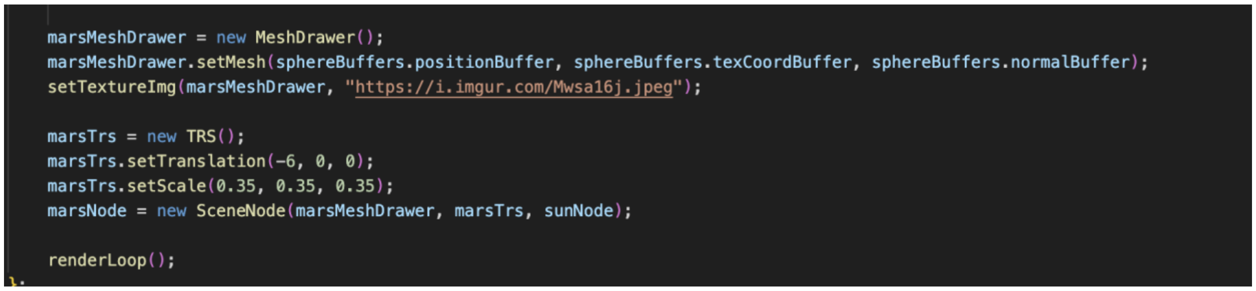
```
//////////////////////////////////////////////////////////////////////////////
// PLEASE DO NOT CHANGE ANYTHING ABOVE !!!
// Calculate the diffuse and specular lighting below.

// Diffuse
float lambertTerm = max(dot(normal, lightdir), 0.0);
diff = lambertTerm;

// Specular
if(lambertTerm > 0.0) {
    vec3 viewDir = normalize(-vPosition);
    vec3 reflectDir = reflect(-lightdir, normal);
    float specAngle = max(dot(reflectDir, viewDir), 0.0);
    spec = pow(specAngle, phongExp);
}

// PLEASE DO NOT CHANGE ANYTHING BELOW !!!
//////////////////////////////////////////////////////////////////////////////
```

Task 3: Adding Mars to the Scene

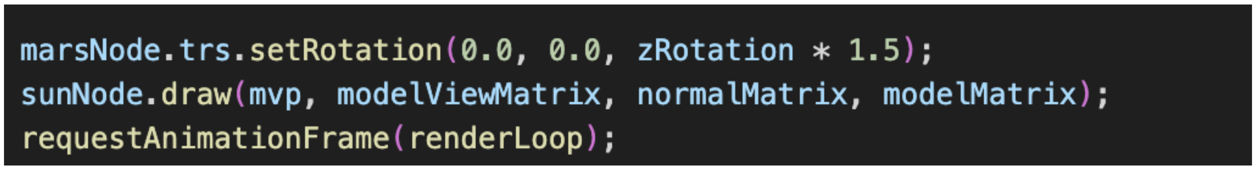
The goal was to add Mars to the solar system, make it a child of the Sun node, and configure its transformations and behavior.

Implementation:

- A MeshDrawer instance was created for Mars and loaded with the sphere mesh and Mars texture (https://i.imgur.com/Mwsa16j.jpeg).
- Mars was translated by −6 units along the X-axis relative to the Sun.
- Mars was scaled uniformly by 0.35 in all dimensions.
- Mars was rotated around its Z-axis at 1.5 times the Sun's rotation speed in the renderLoop.
- Mars was added to the scene graph as a child of the Sun node.



```
    marsMeshDrawer = new MeshDrawer();
    marsMeshDrawer.setMesh(sphereBuffers.positionBuffer, sphereBuffers.texCoordBuffer, sphereBuffers.normalBuffer);
    setTextureImg(marsMeshDrawer, "https://i.imgur.com/Mwsa16j.jpeg");

    marsTrs = new TRS();
    marsTrs.setTranslation(-6, 0, 0);
    marsTrs.setScale(0.35, 0.35, 0.35);
    marsNode = new SceneNode(marsMeshDrawer, marsTrs, sunNode);

    renderLoop();
```



```
marsNode.trs.setRotation(0.0, 0.0, zRotation * 1.5);
sunNode.draw(mvp, modelViewMatrix, normalMatrix, modelMatrix);
requestAnimationFrame(renderLoop);
```