CS 405 Computer Graphics Project Report

Name: Alara Başak
Sabanci ID: 23546

Project Overview

This report outlines the implementation details of the assigned tasks in the CS 405 Computer Graphics course. The tasks focused on manipulating a scene graph, implementing Phong shading in a fragment shader, and enhancing a WebGL-based solar system model with additional functionality.

Task 1: Scene Graph Implementation

Task 1 involved implementing the draw function for the SceneNode class. The goal was to ensure that transformations applied to a parent node are propagated correctly to its child nodes. This was achieved by calculating cumulative transformations combining the node's transformations with those inherited from its parent nodes.
The following code snippet showcases the implementation:

```
  draw(mvp, modelView, normalMatrix, modelMatrix) {

    var transformedMvp = MatrixMult(mvp, this.trs.getTransformationMatrix());
    var transformedModelView = MatrixMult(modelView, this.trs.getTransformationMatrix());
    var transformedNormals = getNormalMatrix(transformedModelView);
    var transformedModel = MatrixMult(modelMatrix, this.trs.getTransformationMatrix());

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

    // Draw children
    this.children.forEach(child => child.draw(transformedMvp, transformedModelView,
transformedNormals, transformedModel));
  }
```

Task 2: Phong Shading in Fragment Shader

In Task 2, the fragment shader was updated to include Phong shading, which comprises ambient, diffuse, and specular lighting components. The implementation involved

calculating the diffuse and specular components within the constraints of the shader's pre-set variables.

```
const meshFS = `...
// Fragment shader code including Phong shading
implementation
`;
  // PLEASE DO NOT CHANGE ANYTHING ABOVE !!!
  // Calculate the diffuse and specular lighting below
  // Diffuse lighting
  diff = max(dot(normal, lightdir), 0.0);

  // Specular lighting
  vec3 viewDir = normalize(-fragPos); // View direction
  vec3 reflectDir = reflect(-lightdir, normal);
  spec = pow(max(dot(viewDir, reflectDir), 0.0), phongExp);

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

Task 3: Enhancing the Solar System Model

The final task was to enhance a WebGL-based solar system model. This included adding rotation to Mars and incorporating Mars as a new object in the scene graph. Mars was scaled and positioned relative to the sun, and a texture was applied to its surface.

Inside Renderloop:

```
        marsNode.trs.setRotation(0, 0, zRotation * 1.5);
```

Inside onload:

```
        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);
```