Sabanci University

CS405 Computer Graphics Course

Project Report 3: Scene Graph + Illumination

Fall 2024-2025

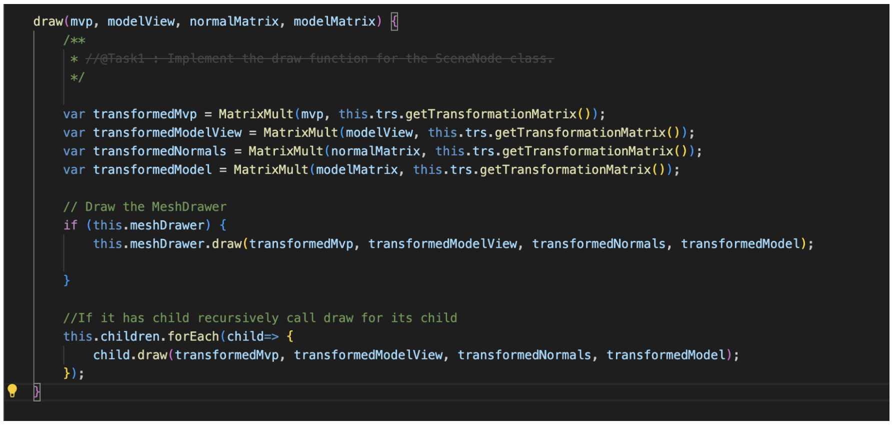
Göktuğ Gökyılmaz
28846

1. Introduction Scene Graph + Illumination

In this project, students completed 3 tasks individually for the solar system model that contains the sun at the center and has two children the Earth and Mars The Earth also has 1 child, the moon

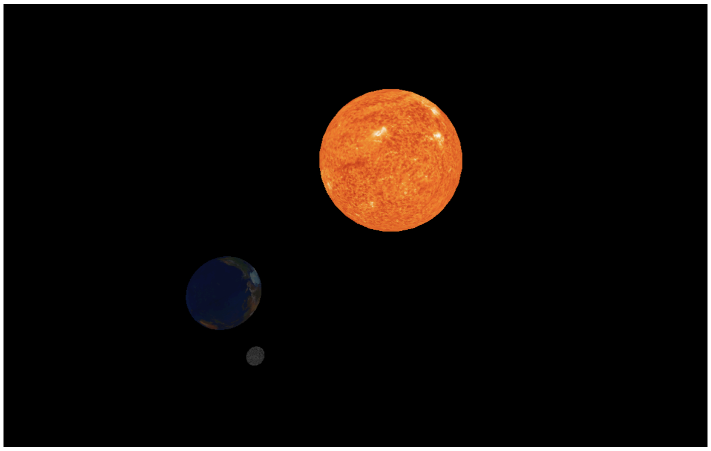
2. Tasks

2.1 Task 1 Implementation of the draw function

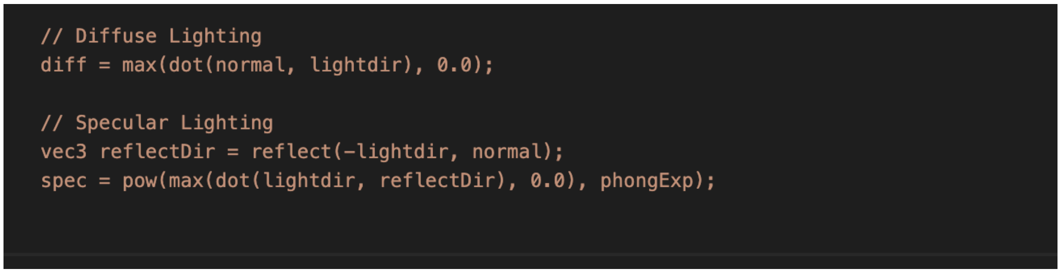


```
draw(mvp, modelView, normalMatrix, modelMatrix) {
    /**
     * ~~//@Task1 : Implement the draw function for the SceneNode class.~~
     */

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

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

    }

    //If it has child recursively call draw for its child
    this.children.forEach(child=> {
        child.draw(transformedMvp, transformedModelView, transformedNormals, transformedModel);
    });
}
```

For the first task, I needed to calculate transformedMvp,transformedModelView, transformedNormals, and transformedModel using the utility function MatrixMult. I calculated those matrixes using this.trs and got the transformation matrix. Then, we used a calculated matrix inside the meshDrawer at the line bottom, and we completed the same procedure recursively for planets' children. After the completion of task 1, get this result

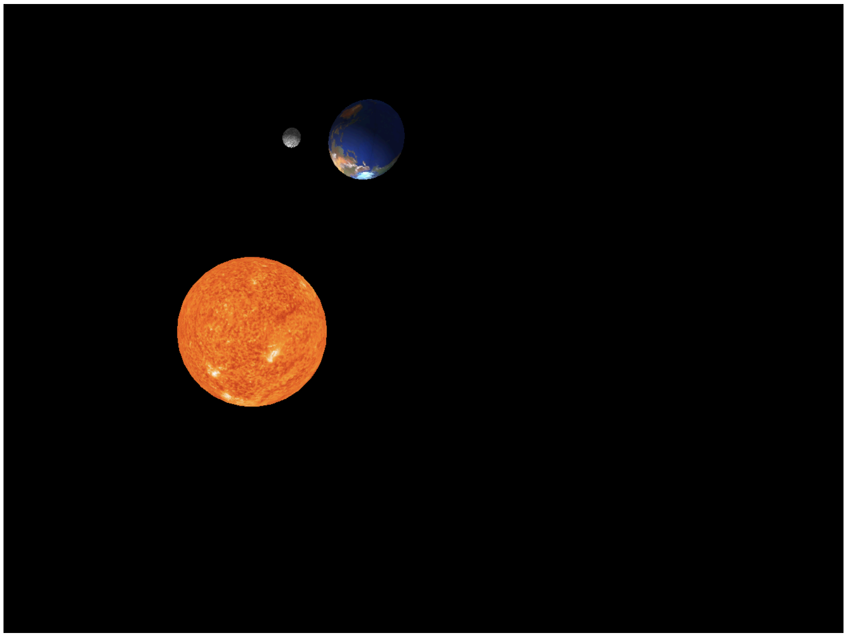


2.2 Task 2 Updating fragment shader inside meshDrawer.js for diffuse and specular light

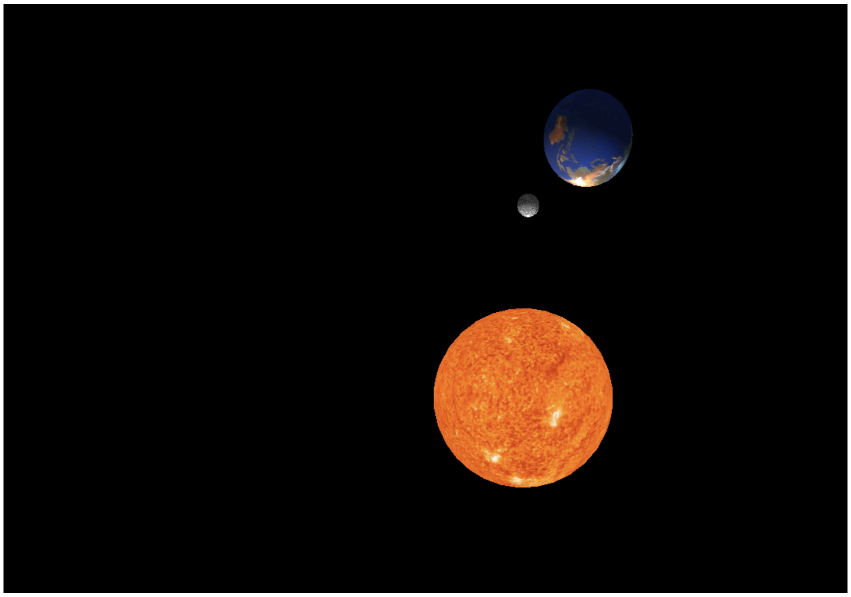


```
// Diffuse Lighting
diff = max(dot(normal, lightdir), 0.0);

// Specular Lighting
vec3 reflectDir = reflect(-lightdir, normal);
spec = pow(max(dot(lightdir, reflectDir), 0.0), phongExp);
```

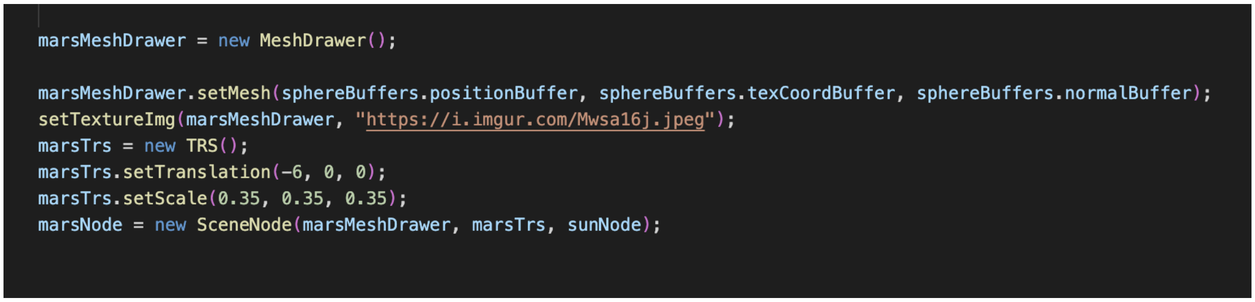
For the second task, we needed to add diffuse and shader light into the fragment shader which is inside meshDrawer.js. I added just 3 lines of code for this task. For the diffusion, I added a single dot product with normal and light direction but we checked that it is always positive by adding a max function which sets 0.0 if the dot product of normal and light direction is negative. For the Specular Light, I got help from the lab 7, for the calculation of reflectDir we used the negative of lightdir which is the direction of light from surface to light direction, not incoming light it is outgoing light when we negate that we got incoming light then calculated reflection direction we used that to calculate the spec by first checking whether is visible or not then use phongExp which is the control of size and sharpness of specular light then use pow function to control its intensity exponentially such as x^y. After this task, I get this diffusion



And after adding specular light I get this

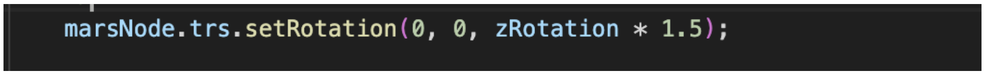


2.3 Task 3 Adding new planet into the solar system



```
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);
```

For this task, we were required to create new MeshDrawer then we set given texture to the new created marsMeshDrawer, Then required translation of x -6, scale it as 0.35 in all axes. To achieve zRotaion of 1.5 we needed to set trs rotation like below.



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

After the all task I achieved this kind of set all planets rotating around sun and mars rotates around the earth.