CS405 Project 3 Report:

In the sceneNode.js we need to update draw function , to draw each node in the scene graph and pass the transformations from parent node to child we use following (From TRS class) to get transformations from parent node.

var nodeTransform = this.trs.getTransformationMatrix();

and,

var transformedModel = MatrixMult(modelMatrix, nodeTransform);

var transformedMvp = MatrixMult(mvp, nodeTransform);

var transformedModelView = MatrixMult(modelView, nodeTransform);

var transformedNormals = MatrixMult(normalMatrix, nodeTransform);

By multiplication, updating the current nodes transformation with the those of its parent. Then, draw each child in loop with corresponding updated variables/matrices.

this.children.forEach(child => {

child.draw(transformedMvp, transformedModelView, transformedNormals, transformedModel);

});

Now, we can see the planets, spinning but we cannot see any lights, so we need to modify the Fragment Shader in MeshDrawer.js .

Firstly, for specular lightning:

* viewDir is the direction from the fragment being shaded towards the viewer (or camera).

vec3 viewDir = normalize(-vPosition);

* Because we want to direction towards the viewer , thus, we negate the -vPosition.
* Direction of the lights reflection calculated with Direction of the light and corresponding normal for fragment.

vec3 reflectDir = reflect(lightdir, normal);

* The last one is the calculation with both reflectDir, viewDir based on Phong Model.

spec = pow(max(dot(viewDir, reflectDir), 0.0), phongExp);

* Finally, we need view direction because specular light is rely on the viewer or cameras position and we need reflect direction to calculate how light is come to camera.

For diffuse lightning:

* diff = max(dot(normal, lightdir), diff);
* We directly calculate diff value with the corresponding normals and the light direction because that’s all we need for diffuse lightning which calculates how light affect the fragment according to its normals.

Finally , we need to add one more planet to the system in, project3.html, renderLoop and onload function.

Firstly, in renderLoop we initialized rotation for the Mars node.

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

A screen shot of a computer

Description automatically generatedSecondly,

A group of planets in space

Description automatically generatedWe initialized mesh, textures and transformations to Mars node, and adding it to a scene nodes to be drawn in meshDrawer with propagated transformations coming from parent Nodes, which is a Sun for Mars.A group of planets in the sky

Description automatically generated