

Task 1

To obtain the transformed matrices for given models, I've multiplied the transformation matrix I obtained using *getTransformationMatrix* method of the *TRS* class with the respective matrices. *TransformedMVP* matrix got formed by multiplying the initial *mvp*, *modelView* matrix got formed by multiplying the initial *modelView* vice versa (Figure 1.1).

```
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());
```

Figure 1.1: Obtaining the transformed matrices.

After obtaining the transformed matrices, I've iterated over them (children) to add it to the scene using the draw function.

Task 2

For this task I've first created a view variable called *cam* that hold the value of the normalized the *vPosition*. I also created another variable *ref* to hold the reflection data which I used the reflect function for. Later, I calculated the specular lighting by taking the dot product of *ref* and *cam* matrices and normalizing it. Specular light also requires *phong* variable to calculate the size of the shiny point on the object, so I added it as well as a power. Finally, I calculated the diffuse lighting by taking the dot product of *normal* and *lightdir* matrices and normalizing it.

```

vec3 cam = normalize(-vPosition);
vec3 ref = reflect(lightdir, normal);

spec = pow(max(dot(ref, cam), 0.0), phongExp);
diff = max(dot(normal, lightdir), 0.0);

```

Figure 2.1: Calculating reflection, position, and spec illumination.

Task 3

For this task, I've edited the html file for the necessary changes requested by the homework document. The changes is as follows:

Inside window.onload function (Figure 3.1):

- I've initialized *marsMeshDrawer* with *positionBuffer*, *texCoordBuffer*, and *normalBuffer* of the *sphereBuffers* class.
- Set the texture image using *setTextureImg* function with *marsMeshDrawer* and the texture imgur URL parameters.
- Created a new transform layer called "*marsTRS*".
- Set the translation to -6 on x, 0 on y, and 0 on z coordinates as requested by the homework document.
- Set the scale as 0.35 on all axes as requested by the homework document.
- Created a new *SceneNode* called *marsSceneNode* with *marsMeshDrawer*, *martTRS* and *sunNode* as parameters.

```

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);
marsSceneNode = new SceneNode(marsMeshDrawer, marsTRS, sunNode);

```

Figure 3.1: The changes inside the window initializer function.

Inside renderLoop function:

- I've added rotation to mars on z-axis as requested. The rotation was added using *setRotation* method of the TRS class by calling it with 0, 0, and $zRotation * 1.5$ parameters.