

# CS 405 Project 3

Kaan Bilgili

## Task 1:

For this task I got the local transformations with the help of TRS class and applied them to MVP, modelview, normal and model matrices.

```
let localTransform = this.trs.getTransformationMatrix();

transformedMvp = MatrixMult(transformedMvp, localTransform);
transformedModelView = MatrixMult(transformedModelView, localTransform);
transformedNormals = MatrixMult(transformedNormals, localTransform);
transformedModel = MatrixMult(transformedModel, localTransform);
```

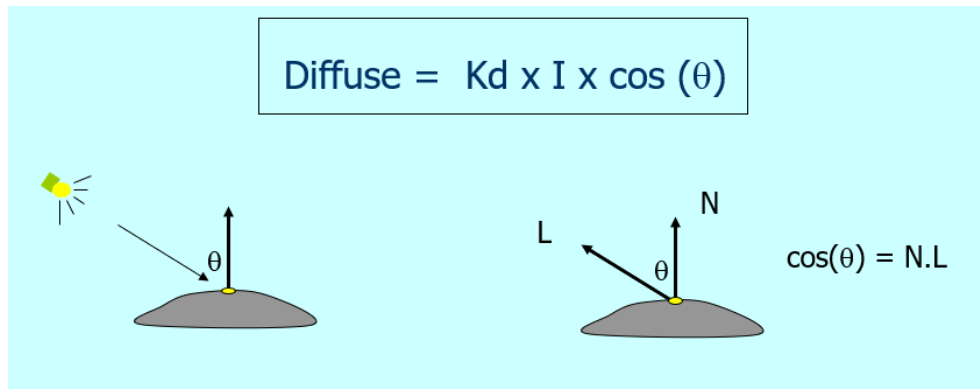
I then recursively applied the transformation I applied to parent to child nodes.

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

This part does applies the transformations recursively. Such that first the transformations are applied to the Sun. Then the draw function is called on the children of the sun (earth and mars after task 3). Then the transformation is applied to the moon as it is the child of earth.

## Task 2:

In this task I first calculated the diffuse lighting as the dot product of light direction vector and normal vector as:



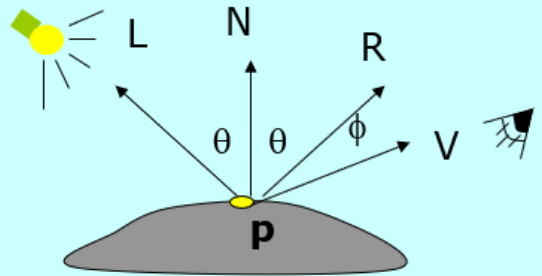
Then I calculated the specular light.

V: vector from P to viewer's eye

R: mirror-reflection direction

$\phi$ : angle between V and R

$\cos(\phi)$ : the larger is  $n$ , the smaller  
is the cos value  
 $\cos(\theta) = R \cdot V$



I first give the view direction as the initialized view direction in the project3.html:

```
camera = new Camera([0, 0, -1], [0, 0, 1], [0, 1, 0]);
```

Then I calculate the mirror reflection direction R. Then I calculate the dot product of R and V to get the cos(theta) value. Then cos(theta) to the power of phong exponential gives us the specular light.

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

vec3 viewDir = vec3(0.0, 0.0, 1.0);
vec3 reflectDir = reflect(lightdir, normal);
float specAngle = max(dot(reflectDir, viewDir), 0.0);
spec = pow(specAngle, phongExp);
```

### Task 3:

In this task I simply copied the earthMeshDrawer and made it marsMeshDrawer. Then I changed the parameters as requested and I put the texture link.

```
marsMeshDrawer.setMesh(sphereBuffers.positionBuffer, sphereBuffers.texCoordBuffer, sphereBu
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);
```

I also added the rotation of mars to the renderloop:

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

At the end of all the tasks I get the following output:

