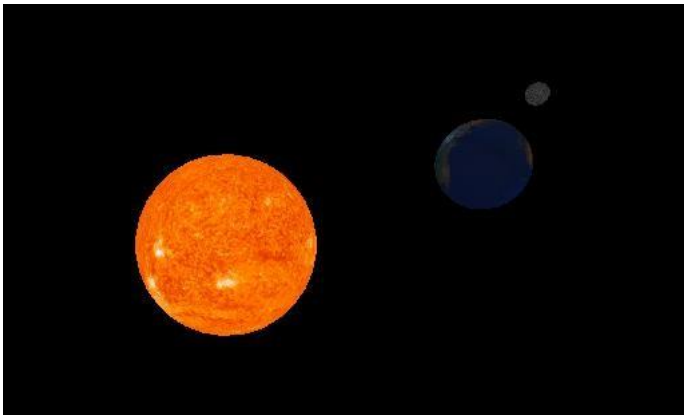
Task 1:

In this section, we use the mvp and modelView variables in the draw function, followed by matrix multiplication with the node's own transformation matrix then we got new varibles. We also obtain the normal variable from the view by using the values obtained from this multiplication to perform getNormalMatrix. Next using modelView matrix, I compute a new normal matrix, which is necessary for accurate lighting and shading in 3D rendering.

If the node has mesh, we determine whether it does,and we use updated matrices to render it.

Lastly, we apply same method recursively to all of current node's child nodes. This guarantees that the parent node's modifications are appropriately inherited by its offspring preserving the scene graph's hierarchical structure and transformations.

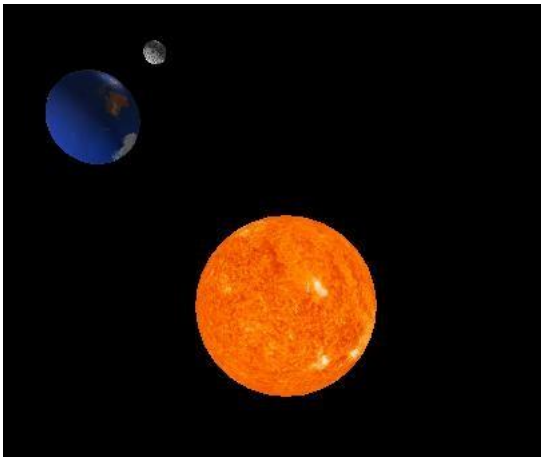
The answer of task 1 is as follows:



Task 2:

For rendering spheres with accurate lighting effects, a full lighting model is used. To make sure that lighting calculations are always correct and shader starts by adjusting normal vector. Then it figures out where the light source is and which way light is going to hit each piece. The shader use diffuse lighting to give scene a general level of light. It figures out diffuse lighting by looking at angle between the light direction and the normal of the surface. This is how light naturally spreads out when it hits surface. To make things look more real, specular lighting is also used to show the bright spots where light bounces back straight at viewer. This is done by comparing the viewer's point of view to the direction of the light that is mirrored. The end color of each fragment is made up of ambient, diffuse, and specular lighting applied to the object's texture. This makes a realistic and appealing spheres.

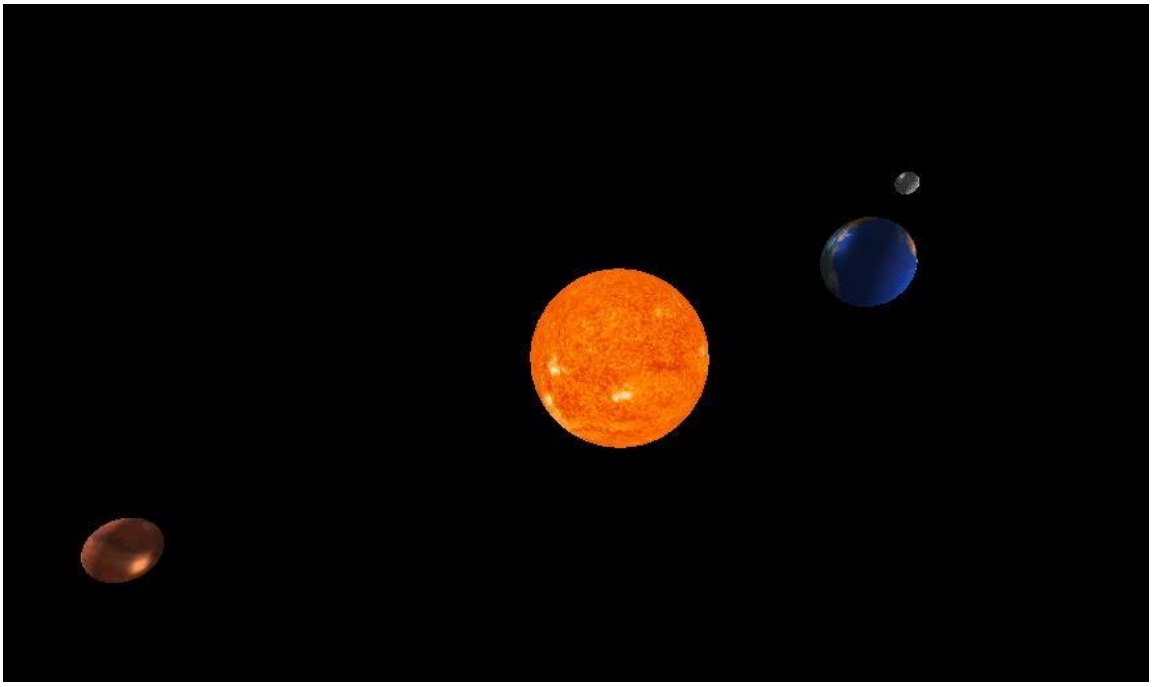
The answer of task 2 is as follows:



Task 3:

In this last task, I first defined a new MeshDrawer and named it marsmesh. Then, I added the necessary spherebuffers values to the marsmesh by doing setmesh. Then I defined a new TRS and set its translation and scale. Then I created a new scenenode with this trMars and marshmesh and connected it to sunNode. Afterwards, I set its rotation with setRotation.

The answer of task 3 is as follows:



: