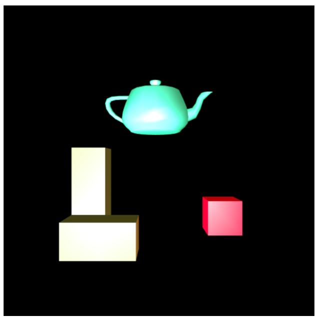
CS 606: Computer Graphics
Report for Assignment 3

Swasti Shreya Mishra (IMT2017043)
International Institute of
Information Technology Bangalore
Email: SwastiShreya.Mishra@iiitb.org



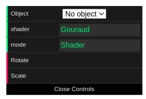


Figure 1. Application window

1. Introduction

The aim of this assignment is threefold—1. To set up lighting for a scene, 2. To experiment with Gouraud and Phong shading models, and 3. To use quaternions for rotation. There is interactive GUI to switch between different modes of the application. Additionally, keyboard events can be used to trigger actions like affine transformations for scene objects. Figure 1 shows the application window.

2. Methods

The entire application is implemented using WebGL and dat.GUI is used for the GUI controls. WebglObjLoader [1] has been used to import mesh data from Wavefront (.obj) files.

2.1. Controls

Figure 2 depicts the controls of the application. The following are the possible operations on the objects depending on the control button:

- **Object**: There are 4 different object selections possible—1. **No object**: This implies that no object is selected and hence we cannot transform the objects, 2. **Teapot**: This will select the teapot object which is at the top of the scene by default, 3. **Cube**: This will select the cube object, and 4. **Ellell**: This will select the leftmost object in the scene by default, which looks like a double L.
- **Mode**: There are two different modes—Shader and Illuminator. We can toggle between these two modes by using the key ‘m’. The default mode is Shader. We can also enable the Illuminator mode by using the key ‘l’.
  1) **Shader**: We can toggle between **Gourad Shader** and **Phong Shader** in this mode by using the key ‘s’. This mode also allows affine transformation of the object is selected in the **Object** field of the controls. If no object is selected, there is no transformation applied to any object.
  2) **Illuminator**: We can switch lights on and off for different meshes in this mode. Depending on which object is selected in the **Object** field of the controls, we can switch the light on for the corresponding mesh by using the key ‘1’ and switch it off by using the key ‘0’. This mode also allows the movement of light inside a constrained bounding box enclosing the mesh object. This movement is constrained only in six directions, i.e. positive and negative X, Y, and Z directions.
- **Rotate**: This performs the operation of rotating the 3 objects by 90 degrees along the x, y and z rotation axis respectively.
- **Scale**: This performs the operation of scaling the 3 objects by 0.5x, 2x, 3x respectively.

2.2. Transformations

To perform transformations on objects/lights, the following keys can be used:

- **Right arrow**: This makes the object/light move by a small step in the +X direction.
- **Left arrow**: This makes the object/light move by a small step in the -X direction.
- **Up arrow**: This makes the object/light move by a small step in the +Y direction.
- **Down arrow**: This makes the object/light move by a small step in the -Y direction.

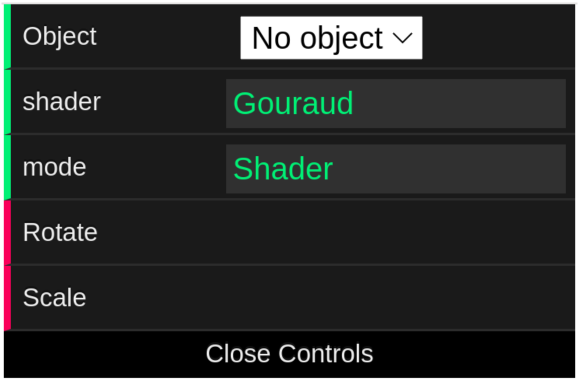


Figure 2. Controls of the application

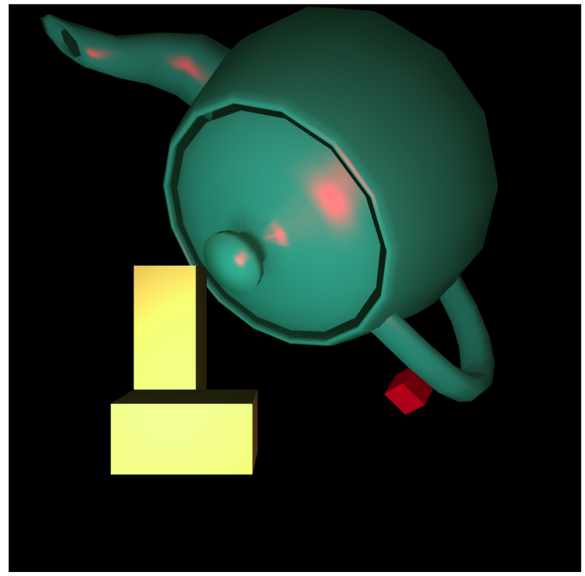


Figure 3. Teapot rotated using trackball quaternion rotation

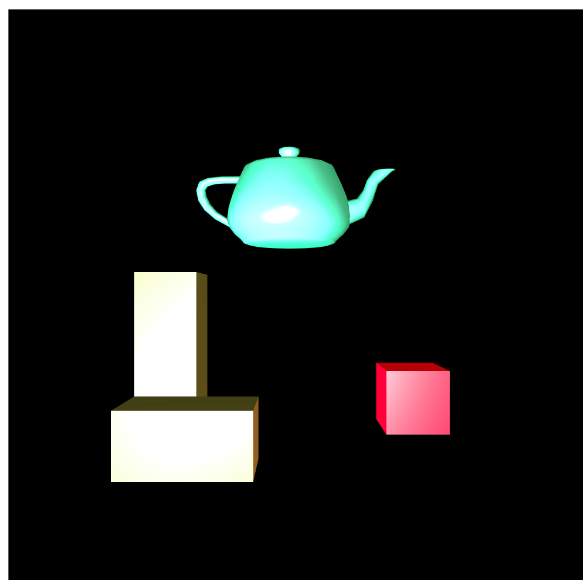


Figure 4. Scene render when constant, linear and quadratic term of attenuation are 1.0, 0.0 and 0.0 respectively

- <: This makes the object/light move by a small step in the -Z direction.
- >: This makes the object/light move by a small step in the +Z direction.
- +: This uniformly upscales the object by 10% (not applicable to light).
- −: This uniformly downscales the object by 10% (not applicable to light).

Note: The movement of light is constrained to a bounding box of the mesh object it corresponds to. This box is 1.25x the size of the bounding box that tightly fits the mesh object. By default, the position of the light is initialised at the top, left and front corner of the bounding box of the mesh object.

2.3. Trackball Rotation

The trackball rotation has been implemented using quaternions of **THREE.js**. Using quaternions we can rotation objects at unusual angles as shown in figure 3. Rotation using quaternions is also preferred as they avoid the gimbal lock situation.

3. Answers To Assignment Questions

3.1. What are your observations of the distance attenuation terms used for lighting?

Attenuation is the phenomenon by which there is a reduction in the intensity of light, which is caused by distance in our lighting. This can be observed in figures 4, 5 and 6. Light becomes weaker the further is travels from its source. This is what attenuation term tries to mimic. Figure 4 shows when light remains constant w.r.t distance. Figure 5 shows when light falls linearly with distance. And figure 6 shows when light falls linearly with distance*distance.

3.2. What are your observations about the change in the shading model?

Figures 7 and 8 show the difference between the rendered image sing both the shaders. The specular lighting is set to red color. We know that, for Gouraud shader lighting equation is used at each vertex, whereas for Phong shader lightning equation is used at each pixel. Similarly, Gouraud shading computes illumination at border vertices and then interpolates it, whereas Phong shading computes illumination at every point of polygon surface. These points align with the fact that Phong shading looks more realistic to the eye compared to Gouraud shading. Also, if we closely look at figure 8, Mach bands are visible, whereas no such bands are visible for figure 7.

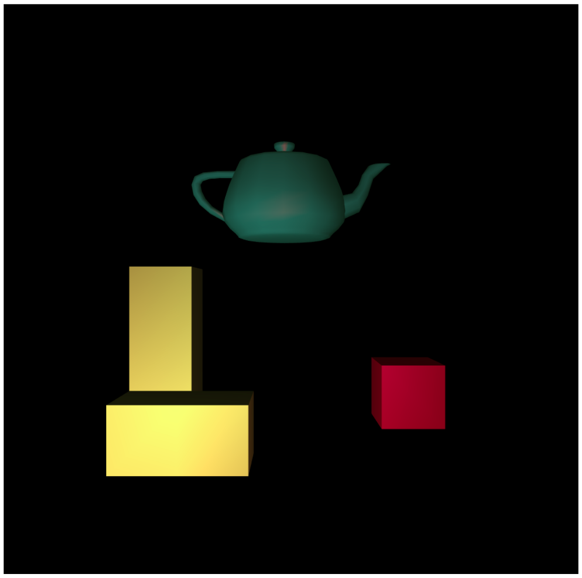


Figure 5. Scene render when constant, linear and quadratic term of attenuation are 0.0, 1.0 and 0.0 respectively

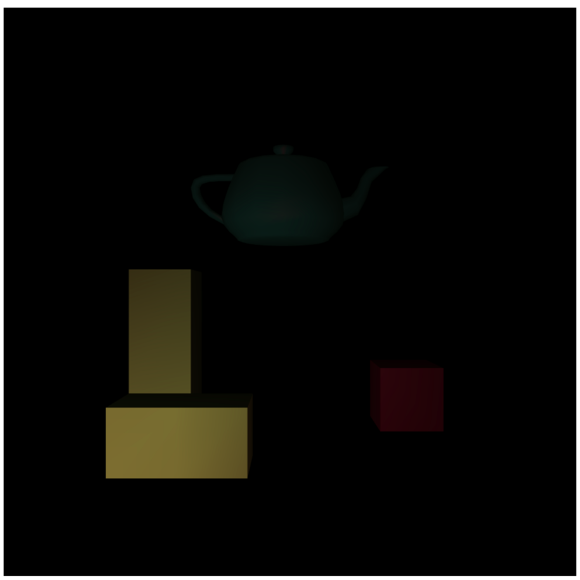


Figure 6. Scene render when constant, linear and quadratic term of attenuation are 0.0, 0.0 and 1.0 respectively

3.3. You are now able to generate different sizes of specular highlights using different settings for lighting, shading, and materials. When do you see focused sharper and smaller specular highlights, and when do you see larger ones?

The specular highlights mainly depend on an object's material. If the material of the object is characterised by high shininess value, the specular highlight is sharper and smaller. The second main thing that determines the pattern of specular highlights is the distance of the light source from the object as well as the way face normals of the object are aligned with the light falling on it. When the object is closer and the face normals are at nearly 180 degree with the direction of light, the specular highlights produced are sharper and smaller. This can be seen in figures 9 and 10.

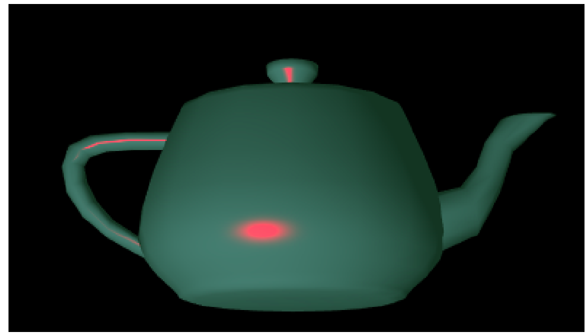


Figure 7. Teapot rendered using Phong Shader

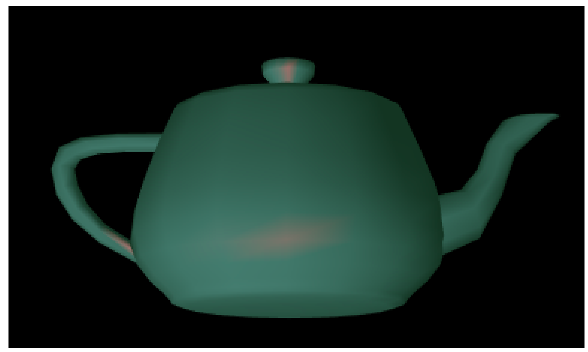


Figure 8. Teapot rendered using Gouraud Shader

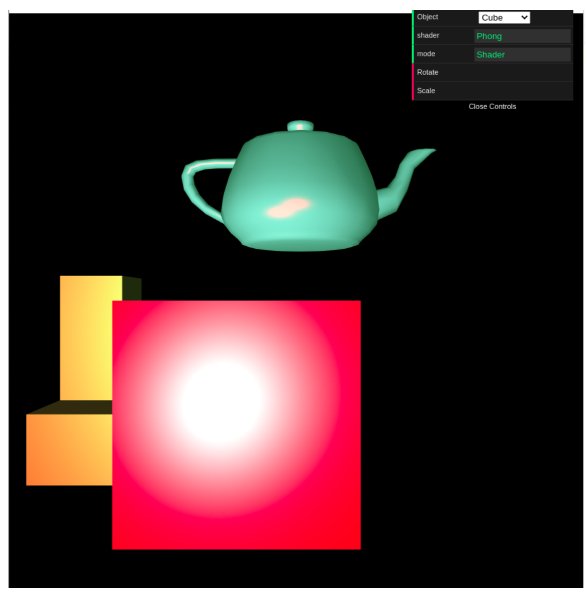


Figure 9. Cube close to the light source, produces sharperand smaller specular highlight

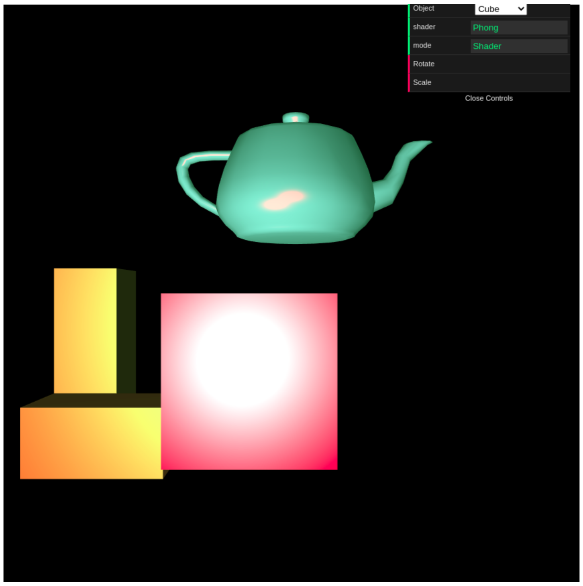


Figure 10. Cube farther from the light source, produces dull specular highlight

3.4. What are your comments about your choice of mesh models for this assignment?

In this assignment, the choices of mesh mattered more than the previous assignment. This is because, to calculate the illumination for an object, we need vertex normals of the object as well. Depending on the geometry of the object the lighting of the objects/scene drastically changes. Additionally, the materials file (.mtl file, generated along with wavefront file while exporting a model from Blender) can be used to set the material properties of the objects.

4. Discussion

The quaternion rotation implemented here has been inspired by the code our TA Amit Tomar provided. This implementation is in THREE.js. Whereas, when I tried a similar implementation using gl-matrix, I faced some issues with scaling. Even though I have understood the concept and the implementation is acceptable according to professors, I will try to implement it using gl-matrix with the help of TAs.

5. Conclusion

This assignment helped us learn illumination in the scene along with phong and gouraud shading and rotation using quaternions. We got to learn about quaternions in depth. Rotation using quaternions is more preferred in computer graphics as they rarely end up in gimbal lock. Therefore, rotation implementation is more robust. We can now render more complex 3D objects, using different illumination techniques as well as add textures on top of it.

References

[1] Aaron Boman. frenchtoast747 / webgl-obj-loader, Apr 2020.