Computer Graphics
Project Report : Illuminator

Prajwal Agarwal (IMT2018056)

1 Problem Statement

Create an application named Illuminator. Create 3-4 objects using blender and import them in webgl using obj format.

(I) Create three to four models and render them on canvas.

(II) Add a point light source corresponding to each model. Use different colors for ambient, diffuse, specular components. Add distance attenuation terms along with reflectance coffecients.

(III) Compute bounding box for each object and ensure that light source is present in area about 1.25 times the size of bounding box.

(IV) Use numeric keys 3 - 6 for selecting mesh models.

(V) Mesh transformation mode : Use key m to perform standard affine transformation. Use arrow keys for translation, +/- for scaling, and mouse dragging for rotation. Rotation needs to be performed using trackball and quaternions.

(VI) Shading Model Choice Mode : Implement Gauraud shading and Phong shading techniques. Use the key s to toggle between them.

(VII) Illuminator mode : Use Key I to set this mode. Use 0/1 to turn off or on the light source. Local illumination model using Bling-Phong Illumination model. Use different keys to manipulate translation of light source.

(VIII) Rotate the camera about x/y/z axis and origin.

2 My approach to the Problem

(I) First I imported only one mesh on the canvas, using the code from Assignment 2. I used the same vertex and fragment shader to start the assignment.

(II) I started implemeting directional lighting using only ambient and diffuse components. I took help from Webgl fundamentals to implement this.

(III) Then I added specular components in shader along with the attenuation equation : $1/ (a + bd + cd^2)$.

(IV) My main goal was to implement a generic Mesh class so that , at the end I can just create multiple objects of that class with different object (.obj) files.

(V) **Gauraud Shading :** In this light of each pixel is calculated in vertex shader. Hence I used uniform variables for ambient, diffuse and specular components. Besides model, view and projection uniform variables, I have used a variable to store the position of light source in the 3D plane. I used the attenuation equation in the vertex shader itself.

(VI) **Phong Shading :** In this light of each fragment in calculated hence most of the computation is done in fragment shader. To implement this, I used the same uniform variables as used in gouraud shading.

(VII) To merge all the contribution of all the four light sources, I used an array of struct to store the components of all light sources. Then I combined them to get the final result.

(VIII) **Bounding Box :** Bounding box is calculated using x, y, z coordinates of vertices. I have scaled the box to 1.25 time its size.

(IX) **Transformation :** I have implemented translation and scaling using translation and scaling matrices as done in previous assignments. The bounding box also moves with the mesh on translation and is being scaled when the mesh object is scaled. The light also moves with same relative position to object when the original object translates.

(X) **Note :** One of the important thing to note is that, if the light position is touching some face of the bounding box, then it will not scale down as that would result in light object going out of the bounding box which is forbidden in this assignment. To scale down the object one needs to first move the light close to the object and then use - to reduce the size.

(XI) **Rotation using Quaternions :** To implement this, I first get the mouse position using event listeners in JS. Then we can convert x-y coordinates to parameteric coordinates on sphere to get the z coordinates. When the mouse moves from some point to another on the sphere, the path which is a part of circle is used to determine rotation of object. We can find axis using cross product and angle using dot product. Now we convert them into quaternion. I have used glMatrix library to do this (kind of black box technique). The main idea here is to combine all small quaternions into a large one that defines the whole rotation. Rotation has been implemented using left mouse click and drag.

(XII) **Toggling shading :** I have used some class variables. One of them stores what type of shading is to be used in the current model. I used toggle between the two shading and re-render the scene. This has been implemented using space bar key.

(XIII) **Illumination Mode :** To turn the lights off, I just use a method to change the specular, ambient and diffuse components to 0. To turn the lights on, I just restore the values using the global states defined.

(XIV) Movement of light is done using a and d for x-axis, w s for y-axis, q and e for z-axis. I just update the position of lighting and re-render the scene. Movement of light is only possible only if the final position is not out of bounding box.

3 Questions and Answers

(I) Effects of distance attenuation terms used for lighting ?
Answer : Light intensity is inversely proportional to the value of ($a + bd + cd^2$). Attenuation simply calculates a percentage of the original light that is used to color a pixel. This light depends distance and keeping the distance same, it will depend more on constant b than on c and light intensity decreases more with the square term. My approach : If the distance is large clearly the constant c needs to be less otherwise light will decrease very rapidly.

(II) Observation about changes in shading models ?
Answer : There were mainly two main observations that I noticed.

When using Phong shading the object appeared to be much smoother than in Gouraud shading. This is evident from the sphere model that I have used which has less segments hence it has corners. In Phong shading the sphere appears to be more smoother. Another observation was that in Gauraud shading the specular point on the model appears a bit distorted. In phong shading it was much more clear, partly because of smooth surface. This is evident from the cube object I have used. Looking at one of the faces, we can clearly observe this.

(III) When do we I see focussed sharper and smaller specular highlights and when do I see larger ones ?
Answer : **Material Reflection :** Changing the value of ambient reflection (k_a) and diffuse reflection (k_d) does not seems to effect the specular highlight. However, when we increase the value of specular reflection (k_s), specular highlight seems to be sharp while decrease leads to vanishing of the specular components. With shininess of the object, when we decrease the shine, the specular highlight spreads over a large area while with high shine, the highlight is sharp, small.
Lighting : With lighting color, the specular highlight actually is of the color of the light rather than color of object.
Shading : In Gauraud shading, we see that the specular highlight is actually larger, a bit distorted while in Phong shading we see a focused sharper specular highlight.

(IV) Comments on choice of mesh models for this assignment ?
Answer : I have tried to take models so that I can clearly display how different types of shading effects the view of models. For example, I have taken a cube so that I can clearly present the specular highlight in both the types of shading implement. To demonstrate that phong shading forms a more smooth image, I have taken a sphere with less segments. All the models serves the purpose of demonstration of lighting with different colors. I tried to make one monkey head metallic to display how the specular highlights forms on such surfaces. I tried to demonstrate how attenuation happens using these models

4 Conclusion

This was a very good starting assignment for me to learn lighting in WebGL. I learnt about different types of shading techniques and how they affect the models on screen. To perform realism, light is necessary, but so is the properties of material itself. I learnt a lot from this assignment.

5 References

1. https://amit-tomar.github.io/T2-20-CS-606
2. YouTube Webgl Tutorial
3. WebGl Fundamentals