Assignment 04 Raytracing

Saumil Lakra, 2021097

Q1.

$$\mathbf{e} + t\mathbf{d} = \mathbf{a} + \beta(\mathbf{b} - \mathbf{a}) + \gamma(\mathbf{c} - \mathbf{a})$$

Make two files namely triangle.cpp and triangle.h. Using ray-triangle intersection equation as given in the slides we calculate alpha, beta, gamma, and A(here with variable name A1) and parameter t. Using Cramer's rule, we get the values required to form the barycentric coordinates of the triangle. We use the following inequality to accept and reject the parameter t.

$$\beta > 0, \gamma > 0, \text{ and } \beta + \gamma < 1$$

Q2.

For this part, we first calculate and set the normal of sphere and triangle and then use the formulas used in Blinn-Phong shading in the previous assignments. We implement the shaders in material.cpp. For the sphere we use the ray position and the position value and store the normal. For triangle we find the normal using the cross product of any 2 adjacent sides and store the normal accordingly. We use the half vector definition here and use the algorithm given on slide 47 to calculate v, l and h.

Q3.

```
Ray shadow(incident.getPosition() + 0.1*l,l);
world -> firstIntersection(shadow);
if(shadow.didHit()==true){
    Color sh = operator*(f,c);
    return sh;
}
```

To implement shadows in the code we put a condition before returning the phong shading. For this we make a new ray i.e. a shadow ray and use the above condition to check when to return only ambient shading. Uncomment the part of shadow in material.cpp.

Q4.

$$\mathbf{r} = \mathbf{d} - 2(\mathbf{d} \cdot \mathbf{n})\mathbf{n}$$

For this part, we use the section of specular reflections(slide 49) and make the shade_ray function in world.cpp recursive. To implement reflections, comment the isSolid->true of sphere in main.cpp. I have taken 20 number of bounces for the light.

Do make the sphere dielectric, I implemented the following algorithm given on slide 58. Do implement the beer law uncomment the required segments in the code.

Sources:

- Code of lab 5
- Code of lab 2
- Lecture 07 of raytracing