Assignment 4 - Report

CSE 333/533

Name:- Aayush Kumar

Roll No.:- 2020008

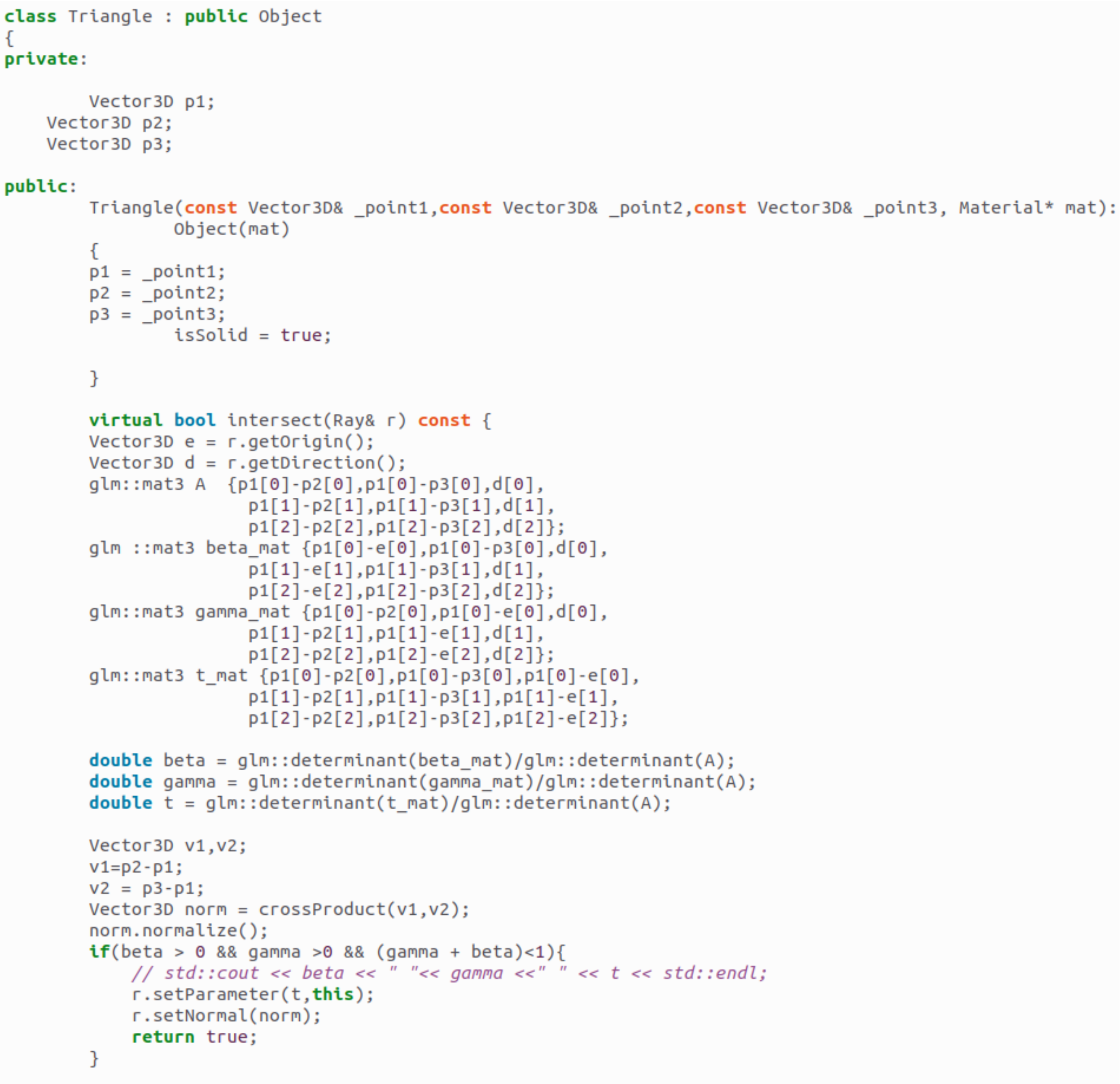
Date:- 16/11/2022

Introduction

This assignment is all about the concepts of Ray Tracing in course CSE 333/533.

Q1

In this question, to implement a triangle, another class triangle.cpp is being used in which the code renders the shape , size of the triangle using the concepts of OpenGL in raytracing. Ray Parameter is used to detect from which direction the ray has hit on which object to detect and show the detection in the output.



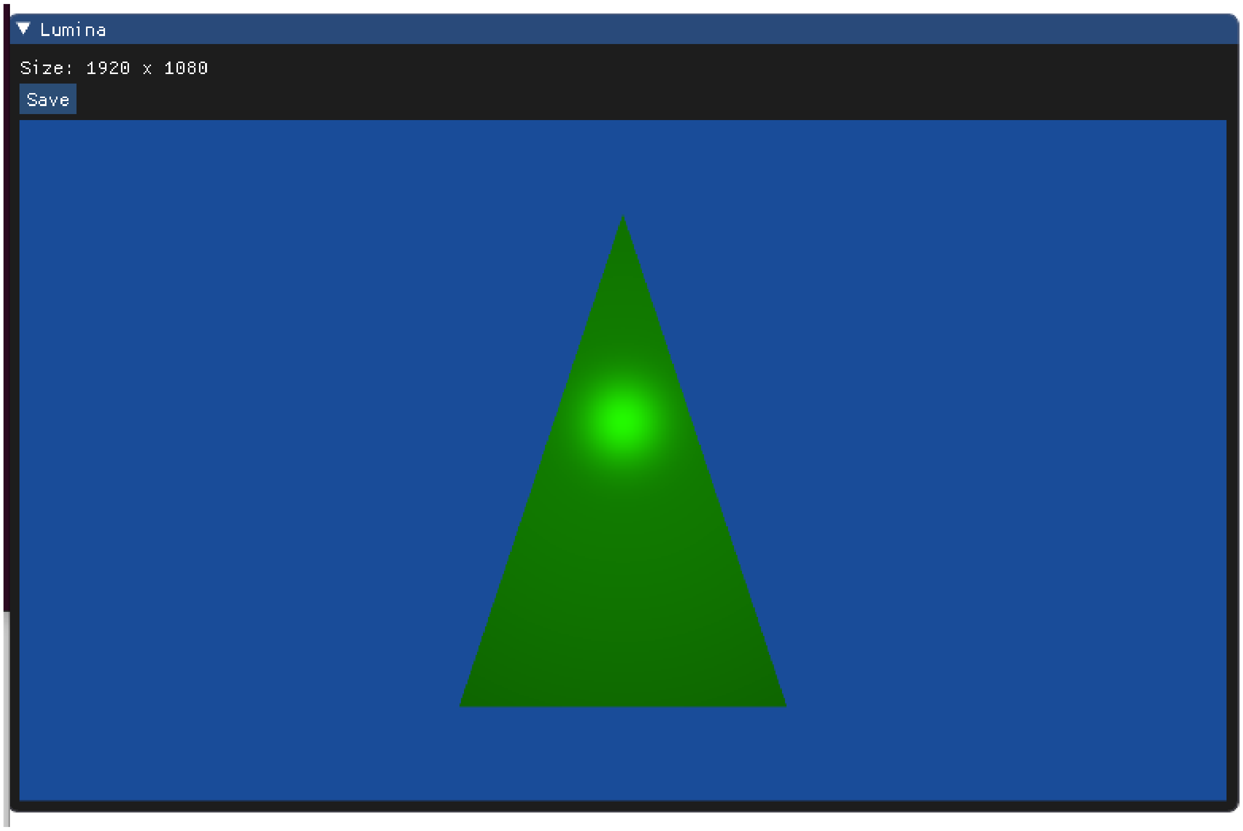
```
class Triangle : public Object
{
private:

        Vector3D p1;
    Vector3D p2;
    Vector3D p3;

public:
        Triangle(const Vector3D& _point1,const Vector3D& _point2,const Vector3D& _point3, Material* mat):
                Object(mat)
        {
        p1 = _point1;
        p2 = _point2;
        p3 = _point3;
                isSolid = true;

        }

        virtual bool intersect(Ray& r) const {
        Vector3D e = r.getOrigin();
        Vector3D d = r.getDirection();
        glm::mat3 A  {p1[0]-p2[0],p1[0]-p3[0],d[0],
                       p1[1]-p2[1],p1[1]-p3[1],d[1],
                       p1[2]-p2[2],p1[2]-p3[2],d[2]};
        glm ::mat3 beta_mat {p1[0]-e[0],p1[0]-p3[0],d[0],
                       p1[1]-e[1],p1[1]-p3[1],d[1],
                       p1[2]-e[2],p1[2]-p3[2],d[2]};
        glm::mat3 gamma_mat {p1[0]-p2[0],p1[0]-e[0],d[0],
                       p1[1]-p2[1],p1[1]-e[1],d[1],
                       p1[2]-p2[2],p1[2]-e[2],d[2]};
        glm::mat3 t_mat {p1[0]-p2[0],p1[0]-p3[0],p1[0]-e[0],
                       p1[1]-p2[1],p1[1]-p3[1],p1[1]-e[1],
                       p1[2]-p2[2],p1[2]-p3[2],p1[2]-e[2]};

        double beta = glm::determinant(beta_mat)/glm::determinant(A);
        double gamma = glm::determinant(gamma_mat)/glm::determinant(A);
        double t = glm::determinant(t_mat)/glm::determinant(A);

        Vector3D v1,v2;
        v1=p2-p1;
        v2 = p3-p1;
        Vector3D norm = crossProduct(v1,v2);
        norm.normalize();
        if(beta > 0 && gamma >0 && (gamma + beta)<1){
            // std::cout << beta << " "<< gamma <<" " << t << std::endl;
            r.setParameter(t,this);
            r.setNormal(norm);
            return true;
        }
```

Some changes were also made in CmakeList.txt to render the triangle at the time of making/compilation of the code.

Q2

Bling - Phong Shading has been done in this assignment. Bling Phong shading consists of three types of shading i.e., Ambient, DIffuse and Specular.
We had initialized the constant values in main.cpp and the object specifications had been added in material.cpp and normalized them with the equation provided in lecture to get the desired output.

The output will be the following for both Q1 and Q2.

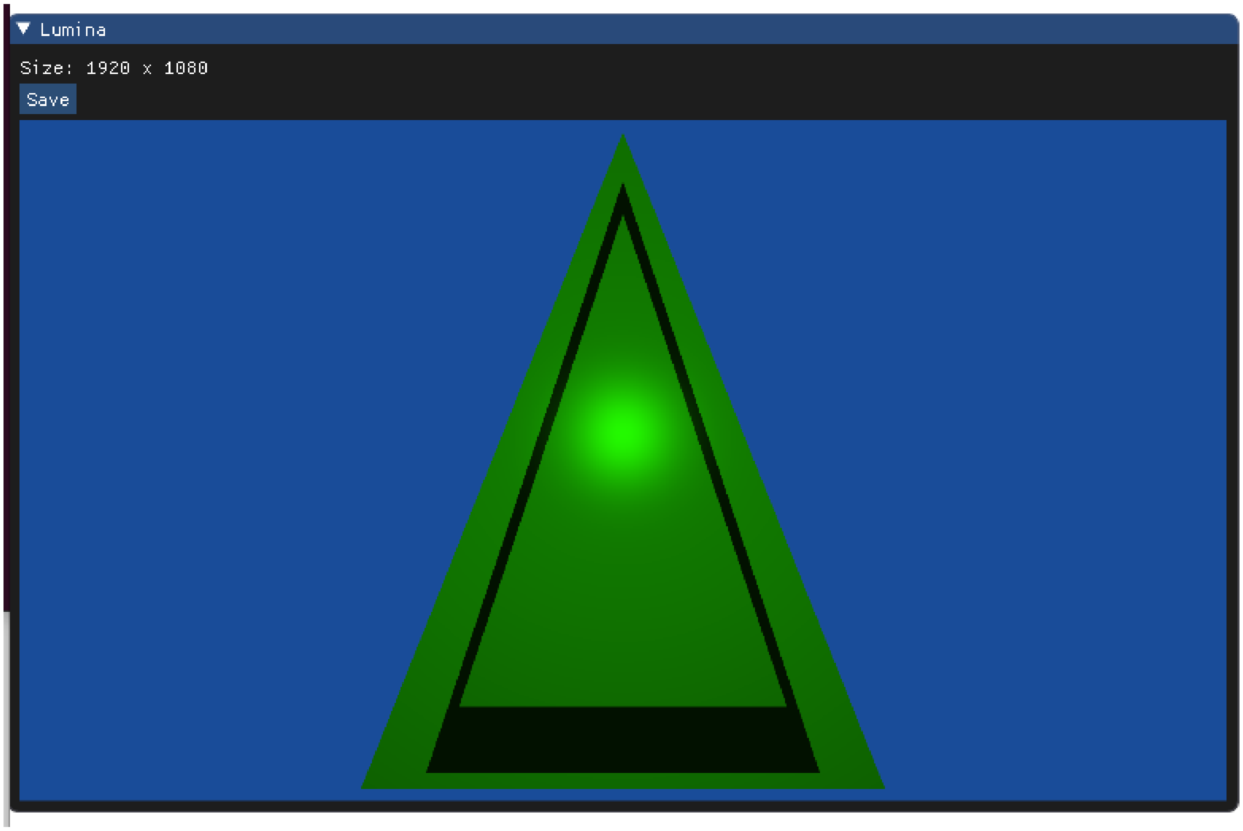


Size: 1920 x 1080

Q3.

For Shadow Implementation , another triangle has been added to show the shadow of one triangle over another. A light source is being added to get the light from a particular direction for the implementation of shadow. If there is any hit of light observed by the object , it will show the ambient and specular lightning; otherwise, diffuse lightning can also be seen from the output.

The resulting output is provided below :-



Size: 1920 x 1080