COSC363 Assignment2 Report

Name: Zheng Chao
ID: 21671773

This project includes five axis-aligned planes and each of the planes have a different colour. The first sphere is reflective, the second refractive, the third transparent and the fourth transparent with n=1.
The default setting is that NUMDIV = 1000, anti-aliasing is not opened and fog is not opened.

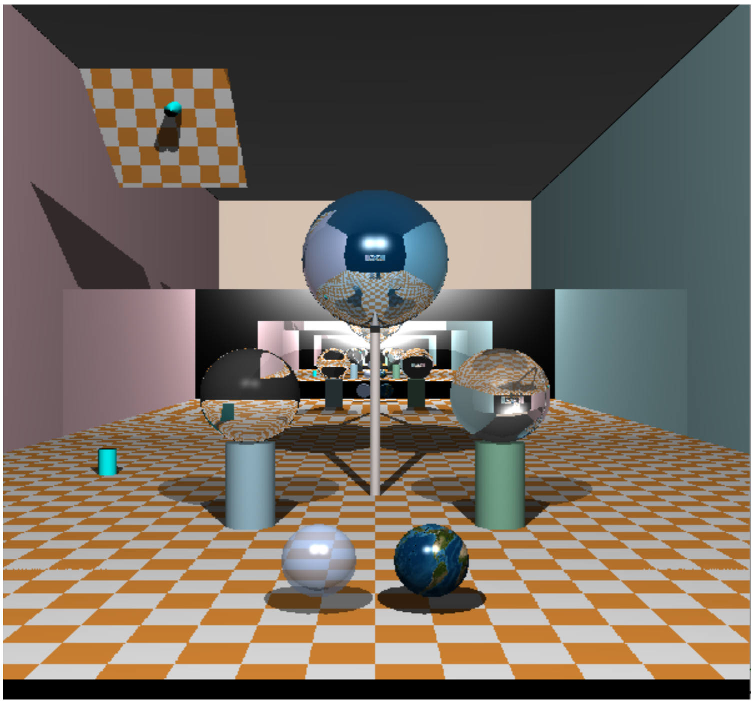
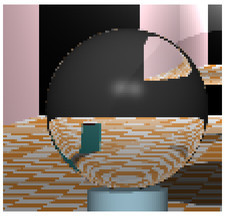
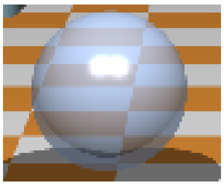


Figure1



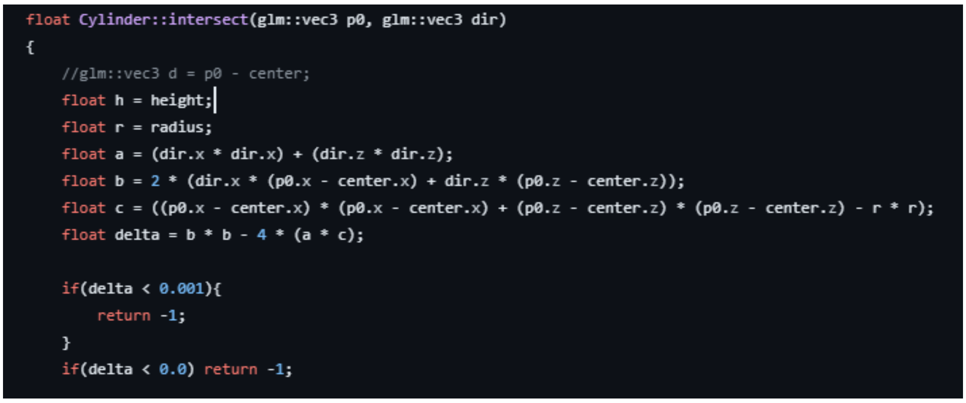
ETA = 1.005



ETA = 1

Extra Features

-Cylinder(with cap)

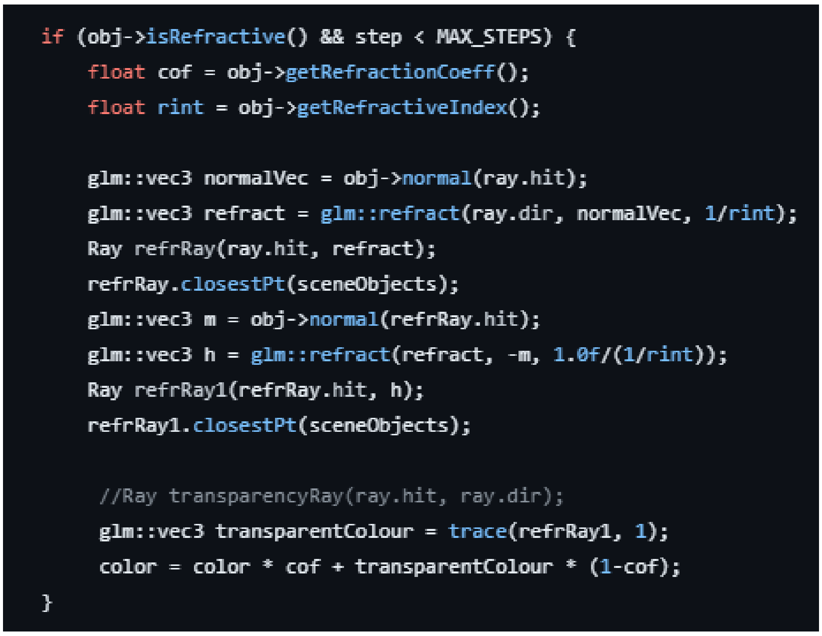


```
float Cylinder::intersect(glm::vec3 p0, glm::vec3 dir)
{
    //glm::vec3 d = p0 - center;
    float h = height;
    float r = radius;
    float a = (dir.x * dir.x) + (dir.z * dir.z);
    float b = 2 * (dir.x * (p0.x - center.x) + dir.z * (p0.z - center.z));
    float c = ((p0.x - center.x) * (p0.x - center.x) + (p0.z - center.z) * (p0.z - center.z) - r * r);
    float delta = b * b - 4 * (a * c);

    if(delta < 0.001){
        return -1;
    }
    if(delta < 0.0) return -1;
```

Figure2

Figure2 shows fixing the intersection equation on a cylinder. To achieve the "cap", I set (center.y + h - p0.y) / dir.y to achieve "cap". Similarly the cap at the bottom can be achieved.

-Refractive sphere

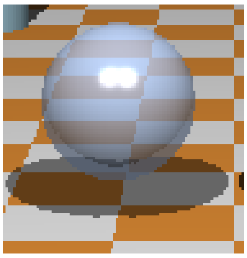


```
if (obj->isRefractive() && step < MAX_STEPS) {
    float cof = obj->getRefractionCoeff();
    float rint = obj->getRefractiveIndex();

    glm::vec3 normalVec = obj->normal(ray.hit);
    glm::vec3 refract = glm::refract(ray.dir, normalVec, 1/rint);
    Ray refrRay(ray.hit, refract);
    refrRay.closestPt(sceneObjects);
    glm::vec3 m = obj->normal(refrRay.hit);
    glm::vec3 h = glm::refract(refract, -m, 1.0f/(1/rint));
    Ray refrRay1(refrRay.hit, h);
    refrRay1.closestPt(sceneObjects);

     //Ray transparencyRay(ray.hit, ray.dir);
     glm::vec3 transparentColour = trace(refrRay1, 1);
     color = color * cof + transparentColour * (1-cof);
}
```

According to page 23 of Lecture 8

-Multiple light sources

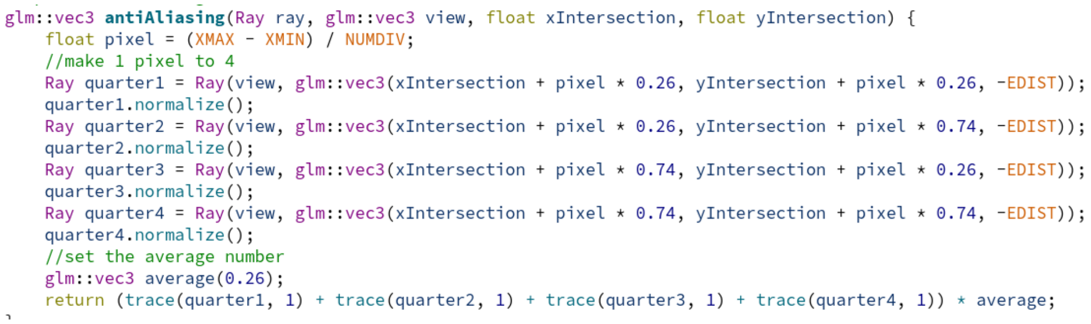
Create a new light.



-Anti-aliasing

Figure6 is what it looks like when anti-aliasing is closed.

Figure7 is what it looks like when anti-aliasing is opened.



```
glm::vec3 antiAliasing(Ray ray, glm::vec3 view, float xIntersection, float yIntersection) {
    float pixel = (XMAX - XMIN) / NUMDIV;
    //make 1 pixel to 4
    Ray quarter1 = Ray(view, glm::vec3(xIntersection + pixel * 0.26, yIntersection + pixel * 0.26, -EDIST));
    quarter1.normalize();
    Ray quarter2 = Ray(view, glm::vec3(xIntersection + pixel * 0.26, yIntersection + pixel * 0.74, -EDIST));
    quarter2.normalize();
    Ray quarter3 = Ray(view, glm::vec3(xIntersection + pixel * 0.74, yIntersection + pixel * 0.26, -EDIST));
    quarter3.normalize();
    Ray quarter4 = Ray(view, glm::vec3(xIntersection + pixel * 0.74, yIntersection + pixel * 0.74, -EDIST));
    quarter4.normalize();
    //set the average number
    glm::vec3 average(0.26);
    return (trace(quarter1, 1) + trace(quarter2, 1) + trace(quarter3, 1) + trace(quarter4, 1)) * average;
}
```

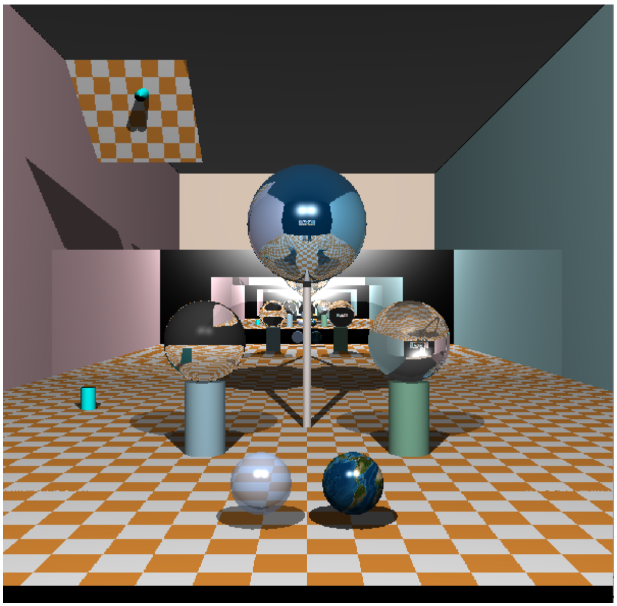


Figure6

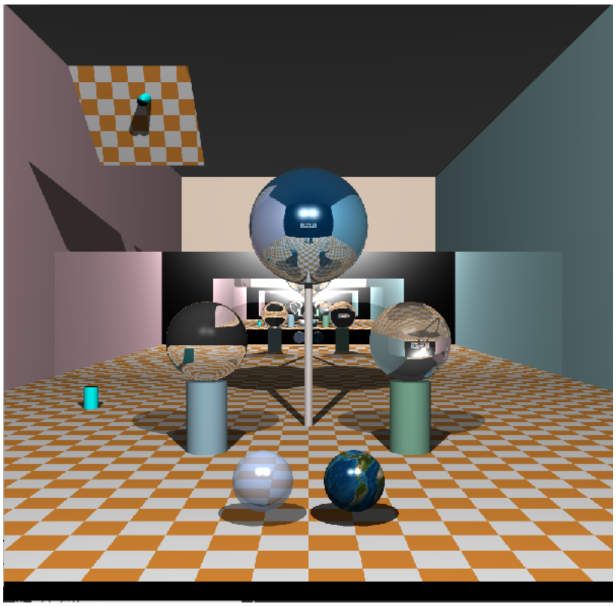
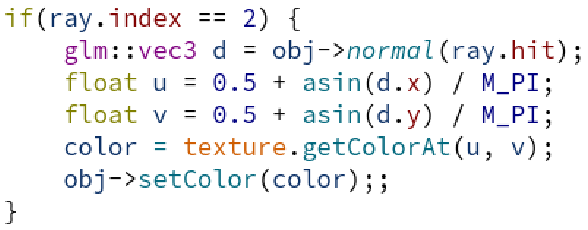


Figure 7

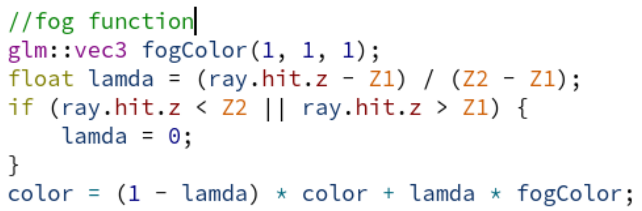
-Textured sphere



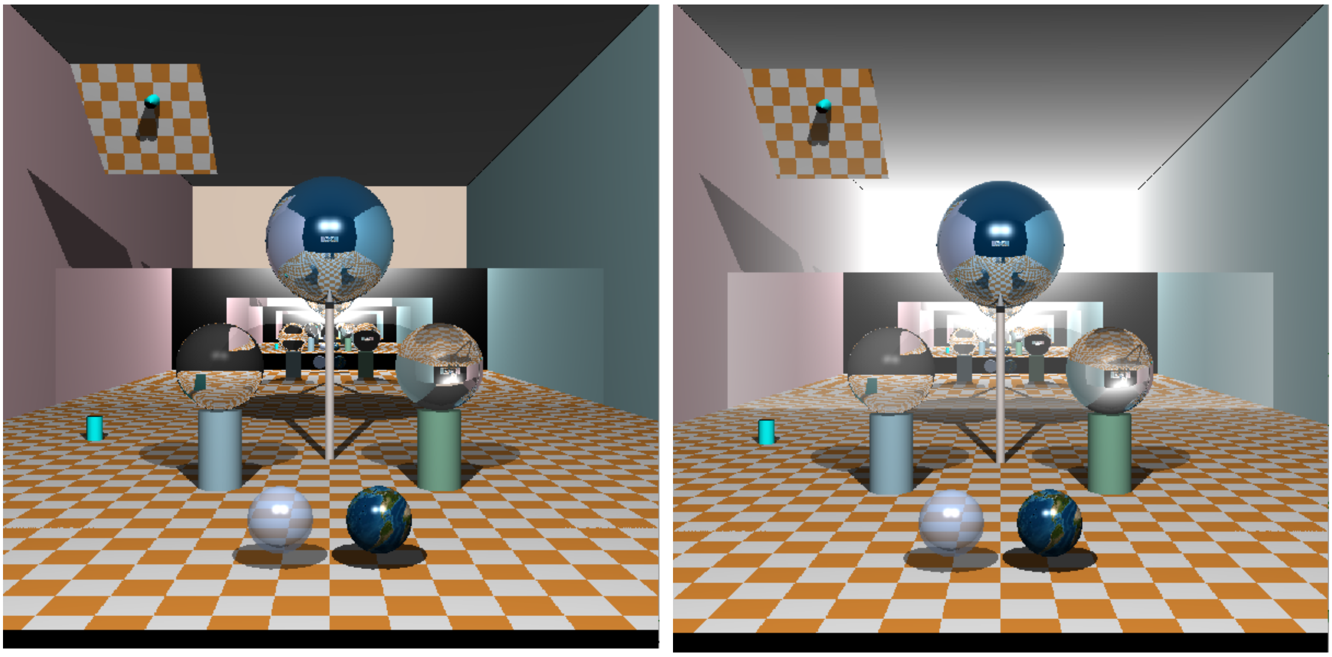
```
if(ray.index == 2) {
    glm::vec3 d = obj->normal(ray.hit);
    float u = 0.5 + asin(d.x) / M_PI;
    float v = 0.5 + asin(d.y) / M_PI;
    color = texture.getColorAt(u, v);
    obj->setColor(color);;
}
```

-Fog

set z1 = -100 and z2 = -300;



```
//fog function
glm::vec3 fogColor(1, 1, 1);
float lamda = (ray.hit.z - Z1) / (Z2 - Z1);
if (ray.hit.z < Z2 || ray.hit.z > Z1) {
    lamda = 0;
}
color = (1 - lamda) * color + lamda * fogColor;
```



-Successes and failures.

I have implemented as much basic functionality as possible while accommodating multiple light sources and multiple mirrors.

The failure is that with multiple mirrors, the background colour of the reflections is confusing and the effect of the light source causes some white areas in the mirrors and affects the general aesthetics.

References

Earth Observation Research Center, Japan Aerospace Exploration Agency. (2005, June 30). Mount Everest and the Himalayas. Retrieved March 31, 2023, from https://www.eorc.jaxa.jp/en/earthview/2005/tp050630.html