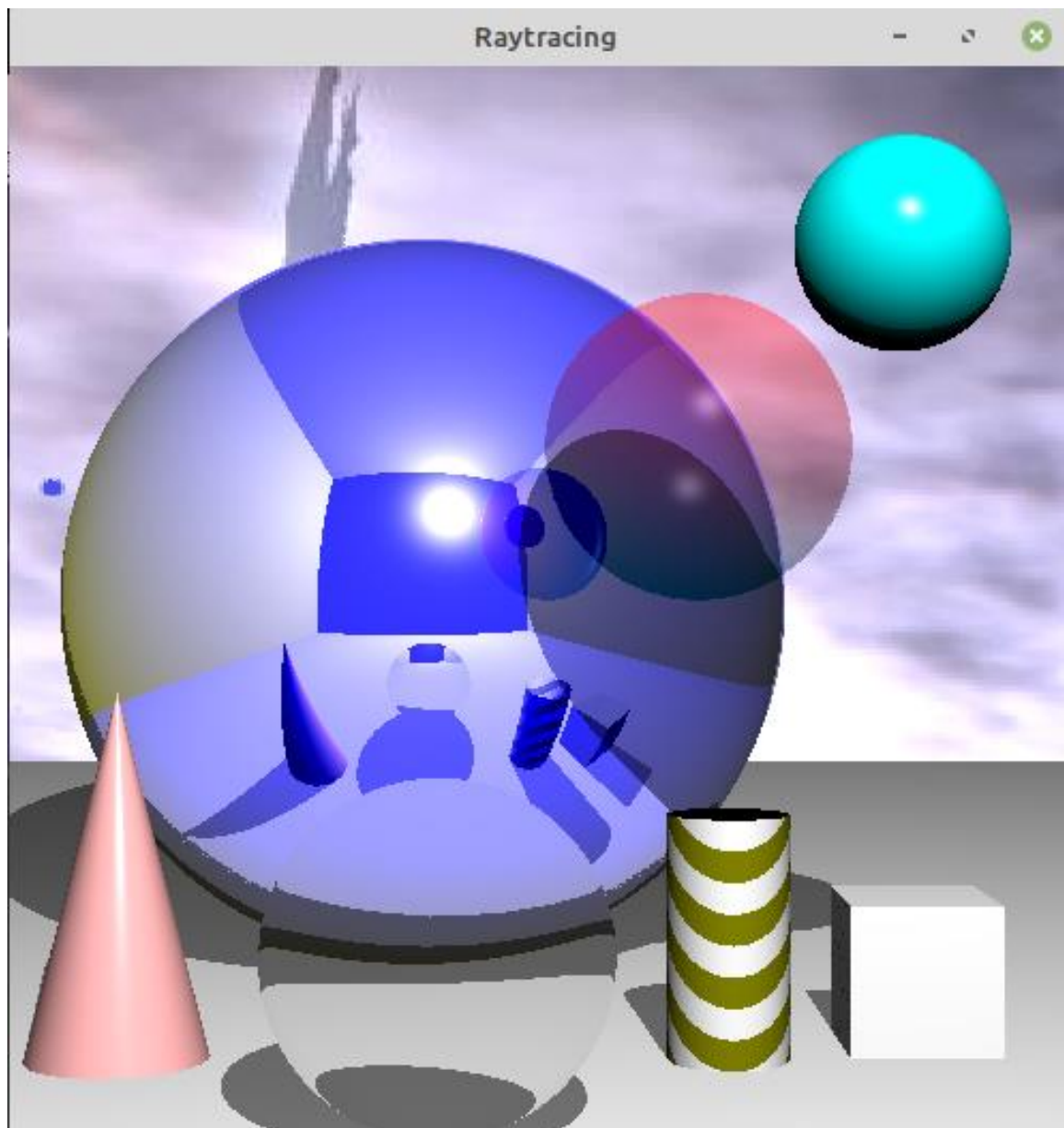


Chenying You  
cyo23

## COSC363 Assignment 2 Report

**Build Command :** `g++ -Wall -o "%e" RayTracer.cpp Ray.cpp SceneObject.cpp  
Sphere.cpp Plane.cpp Cylinder.cpp Cone.cpp TextureBMP.cpp -lm -lGL -lGLU -lglut`



### Description:

This scene contains 4 spheres, one cube, one cylinder and one cone. One fixed light source, and a box enclosed this objects with a plane. The big blue sphere is transparent and reflective, while one sphere is refractive.

Extra Features:

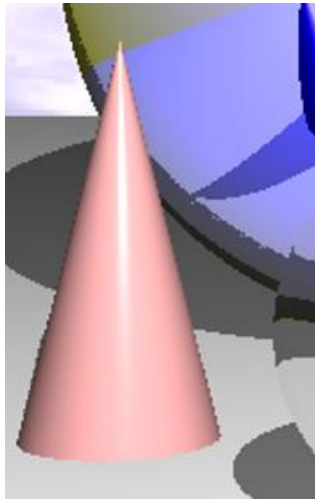
Cylinder and A non-planer object textured using a procedural patter:



```
Cylinder *cylinder = new Cylinder(glm::vec3(6, -15, -70), 2, 8.0, glm::vec3(1, 0, 0));
sceneObjects.push_back(cylinder);

if (ray.index == 4) {
    if ((int(ray.hit.y+ray.hit.z-4)) % 2 == 1 || int(ray.hit.y+ray.hit.z-4)%2 == -1) {
        color = glm::vec3(1, 1, 1);
    }
    else {color = glm::vec3(0.5, 0.5, 0);
    }
    obj->setColor(color);
}
```

Cone:



```
Cone *cone = new Cone(glm::vec3(-14, -15.0, -70.0), 3, 12.0, glm::vec3(1, 0.7, 0.7));
sceneObjects.push_back(cone); //index5
```

Anti-aliasing:

As in my code part:

```

glm::vec3 anti_alias(float xp, float yp, float cellX, float cellY, glm::vec3 eye, int step)

    //list<glm::vec3> colors;
    glm::vec3 colors[4];
    for (int i = 0; i < 4; i++) {
        glm::vec3 newDir(xp + 0.25 * cellX, yp + 0.25 * cellY, -EDIST);
        if (i == 1) {
            glm::vec3 newDir(xp + 0.25 * cellX, yp + 0.75 * cellY, -EDIST);
        }
    }

```

Refraction:

