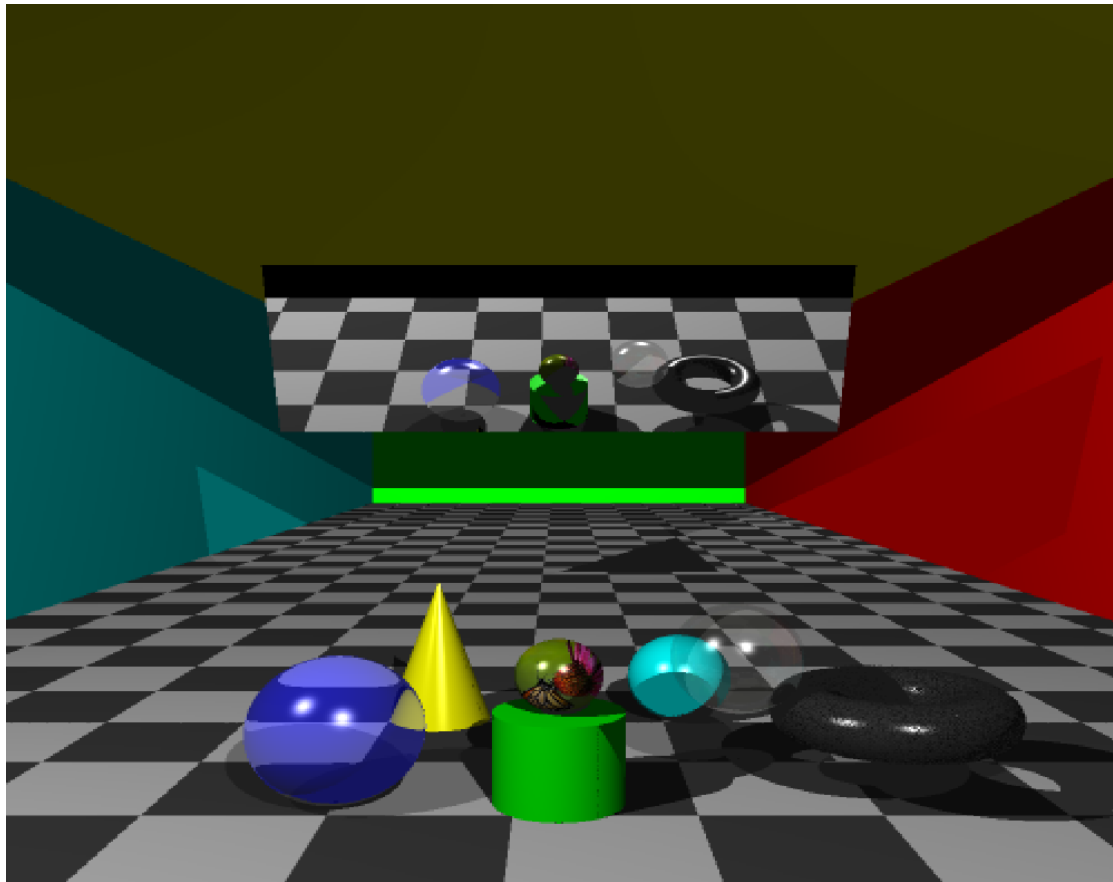


COSC363 Assignment 2 – Ray Tracer

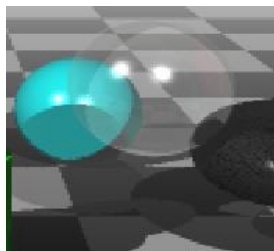
Carl Chen (82830054)



The scene includes a 5 side box room with several objects including a sphere, cone, cylinder, and torus. The scene takes about 15secs to render with antialiasing and 7 sec without.

Basic features:

- a) 5 sided box with different colours for each side with objects arranged closer to the camera.



- b) A transparent sphere
- c) Object shadows, lighter shadows for transparent and refractive objects
- d) A mirror plane towards the top back of the scene facing down to see the back of the objects.
- e) The floor plane has a chequered pattern.

Extra features:

- a) Other objects:

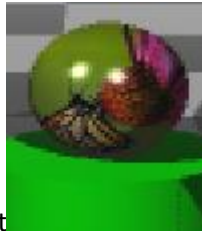
- a. Cone: Constructed using coordinates of the centre of the base of the cone, radius and height. Surface normal is calculated using

```
float r = sqrt((p.x-center.x)*(p.x-center.x) + (p.z-center.z)*(p.z-center.z));
glm::vec3 n = glm::vec3(p.x-center.x, r*(radius/height), p.z-center.z);
n = glm::normalize(n);
```

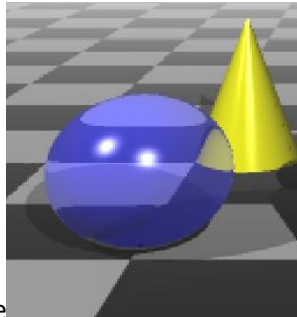
- b. Cylinder: Constructed using coordinates of the base centre, height and radius. Surface normal is calculated using

```
glm::vec3 n = glm::vec3(p.x - center.x, 0, p.z - center.z);
n = glm::normalize(n);
```

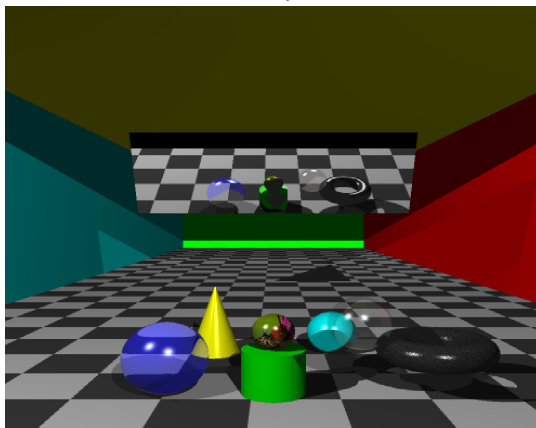
- c. Torus: Constructed using a spherical bounding volume and a quadratic equation solved by code provided enrich666 from Git Hub.



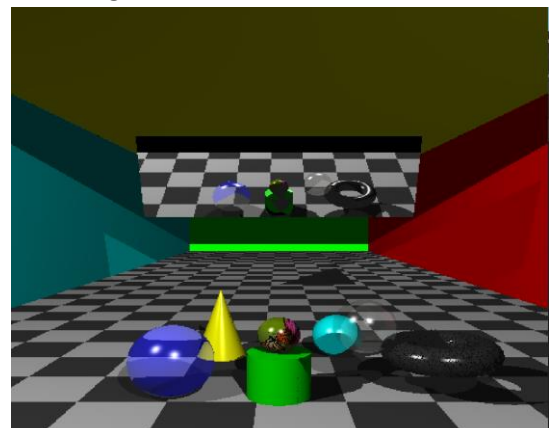
- b) Visible cap for the cylinder object



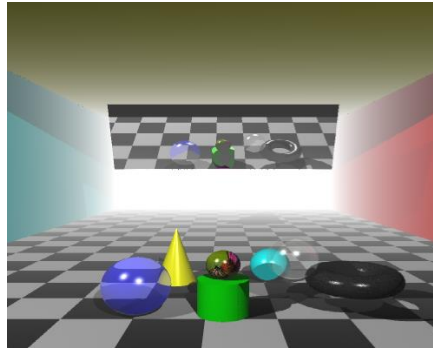
- c) Refractive sphere
d) Two light sources and specular highlights and shadows
e) Anti-aliasing: The scene takes 27sec to render with anti-aliasing and 7secs without. It's purpose is to reduce aliasing artifacts by taking multiple samples within each pixel and averaging the results to produce smoother and more accurate images, especially along the edges. The function calculates four sample directions within the pixel and then offsets the centre of the pixel by small amounts to get four different directions. Then four rays are cast, and results are averaged.



With AA



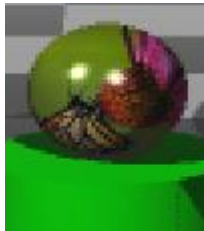
Without AA



f) Fog

```
glm::vec3 n = obj->normal(ray.hit);
double tu = asin(n.x) / M_PI + 0.5;
double tv = asin(n.y) / M_PI + 0.5;
obj->setColor(texture.getColorAt(tu, tv));
```

g) Sphere texture:



Declaration

I declare that this assignment submission represents my own work (except for allowed material provided in the course), and that ideas or extracts from other sources are properly acknowledged in the report. I have not allowed anyone to copy my work with the intention of passing it off as their own work.

Name: Carl Chen

Student ID: 82830054

Date: 03/06/2024