COSC363 – Assignment 2
Ray Tracer

Louis Davies
Lda62
94941666

Report:

Ray Tracer:
The project is a ray tracing scene that consists of a reflective, refractive, transparent and textured sphere, a cone, and a cylinder on a chequered plane surface with a procedurally generated patterned background.

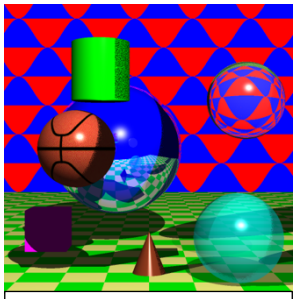

Fig 1: Scene

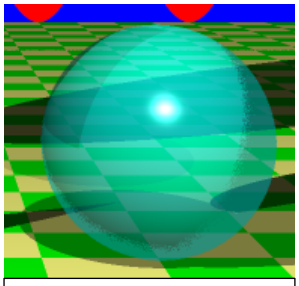

Fig 2: Transparent sphere

Transparent Object:
The turquoise sphere is a transparent object, this was achieved by sending another ray in the same direction as the to get the colour of what is behind it. As you can see in figure two the shadow is lighter than that of the solid objects, this was achieved by first testing if object casting the shadow is transparent or refractive and then multiplying the colour value by a value calculated using the transparency or refractive coefficient.

Cube Constructed From Planes:
The purple cube was constructed from six planes and casts appropriate shadows from both light sources. One light source is above the scene near the middle and the other is above the scene to the far left as you can see by looking at the shadows produced by the objects in the scene.

Chequered Pattern on Plane:
The chequered pattern was produced by building upon on the stripe pattern from the lecture notes to check both the z and x values.

Extra Shapes Cone and Cylinder:
Using the formulae given in the lecture notes cone and cylinder objects with there own intersect and normal methods were created and added to the scene. However, these objects seemed to have extra pixelated dark self-casted shadows on the side opposite a light source which I was unable to find a fix for.

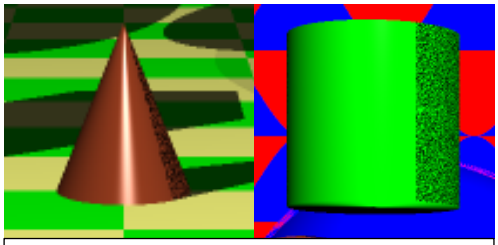

Fig 3: Cone and cylinder

Refractive Sphere:
Refraction was implemented by testing the objects isRefractive method in the trace function. Then the colour is obtained by getting the colour from the point where the refracted ray ends up. The refracted ray and its direction are obtained using the formula from the lecture notes. The sphere shown has a refractive index of 1.1 and a refractive coefficient of 0.8. Its shadow is lighter than the solid objects and the transparent sphere as it has a higher coefficient than the transparent sphere.

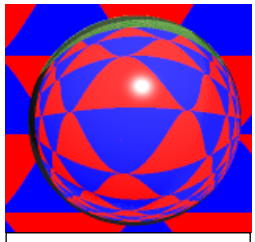

Fig 4: Refractive sphere

Multiple Light Sources:
There are two light sources one above the scene slightly off centre and one to the far left of the scene these both produce shadows as seen in fig one.

Anti-Aliasing:
This was implemented in the display function in order to smooth out colour transitions for a better looking scene. Instead of creating one ray in the centre of each cell four were created and traced in the four quarters of each cell and the colour of the cell is determined by average rgb of the four cells. However this greatly increased the load time of the scene.

Non-Planar Textured Object:
For this aspect of the scene a sphere was texture using a basketball bmp from carloschapeton.com this is shown in figure five. The s and t values were computed by using the following formulae where the vector in the used in the normalize function is the centre of the sphere.

```
glm::vec3 norm = glm::normalize(ray.hit -
glm::vec3(-7.5, 0, -60));
float s = atan2(norm.z, norm.x)/(2*M_PI) + 0.5;
float t = asin(norm.y)/M_PI + 0.5;
```

Fig 5: Textured Sphere

Procedural Pattern:
The procedural pattern generated on the back plane is generated using the sine function the pattern in the scene uses an amplitude of five. It multiplies the current x value by 10/pi then gets the value of the current y value mod (amplitude x 4) and if it less than amplitude x 2 it offsets the wave by pi. It then gets the value of amplitude * sin(altered x value) + amplitude to get the y value of a sine wave sitting on the x axis (y range 0-2*amplitude). It then checks if the current y value mod (2*amplitude) is above or below the computed sine wave y value and sets the colour accordingly. I also implemented my own recursive float mod function in order to do this with floats so to get a much smoother curve.

Time to Generate:
The time to generate the scene is approximately 10 seconds. This is quite long considering the expectations of computer applications today, this long load time can be contributed partly to a slow virtual machine, which ran on a delay of roughly 100ms during basic operation and inefficient algorithms for anti-aliasing and ray tracing.

Build instructions:
Unzip file and build project from cmakelists file and run RayTracer.out.

References:
Basketball Texture: http://www.carloschapeton.com/workstuff/simpleMan/textures/