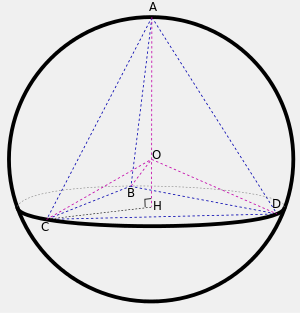
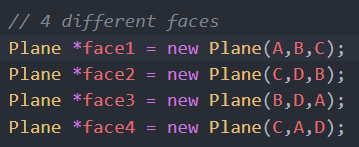
**COSC363 Assignment 2 Report**

**Build Command:** **Linux** = cd Linux && make, **Windows** = cd Windows && make

**Features:**

1. **Tetrahedron**

* The Tetrahedron is drawn as shown

Figure 1- https://en.wikipedia.org/wiki/Tetrahedron

* Constructed 4 faces using the plane class.

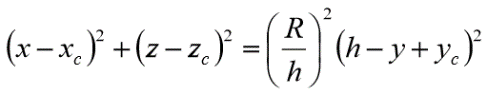
**Ray equation​**

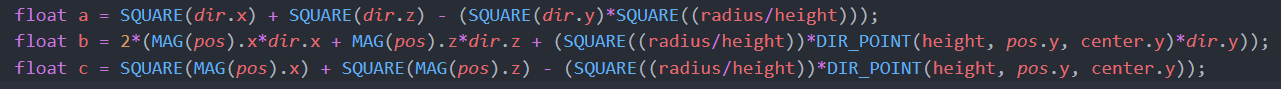
* Sub Ray equation(s) in to intersection equation to find roots.
* Related to specific object: d = direction, o = origin, c = centre, t = distance from ray’s origin to point of ray

1. **Cone**

* Cone class was created in *Cone.h* with a *intersect* and a *normal* method.
* Normal Method:

Normalize’s the given point and returns a unit normal vector

* Intersect Method:

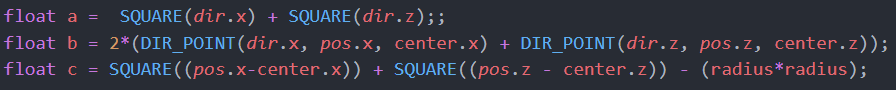
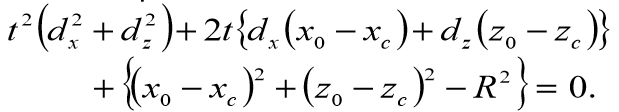
Contains the intersection equation of the cone object and finds the roots below

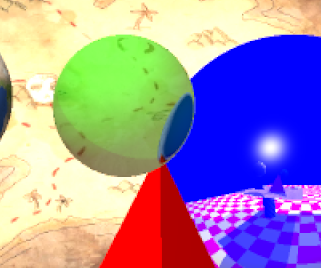
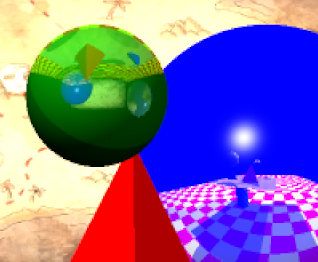
1. **Cylinder**

* Cylinder class was created in *Cylinder.h* with a *intersect* and a *normal* method.
* Normal Method:

Normalize’s the given point and returns a unit normal vector

* Intersect Method:

Contains the intersection equation of the cone object and finds the roots below

****

1. **Refraction and Transparency**

* Algorithm:

ray is traced twice, need 2 normals and refractive rays

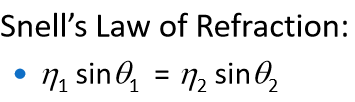
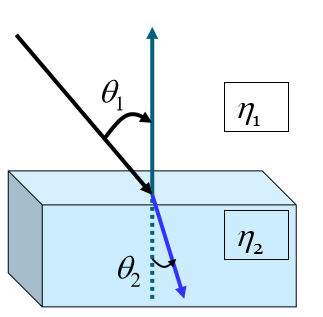
* This is described using Snell’s Law

Figure 4.1 – Taken from lecture 8, slide 19

Figure 3: ETA=1.6

Refraction

Figure 2: ETA=1.003

Transparent

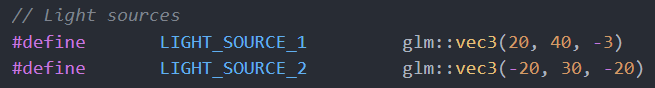
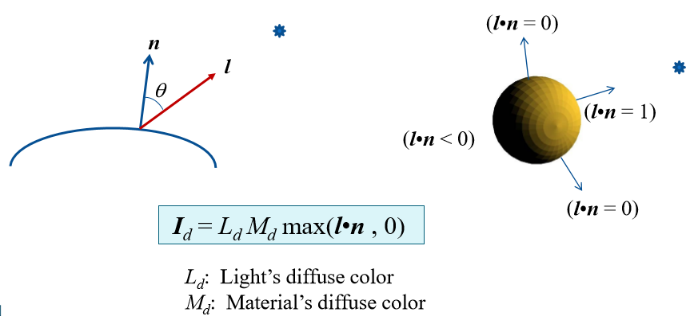
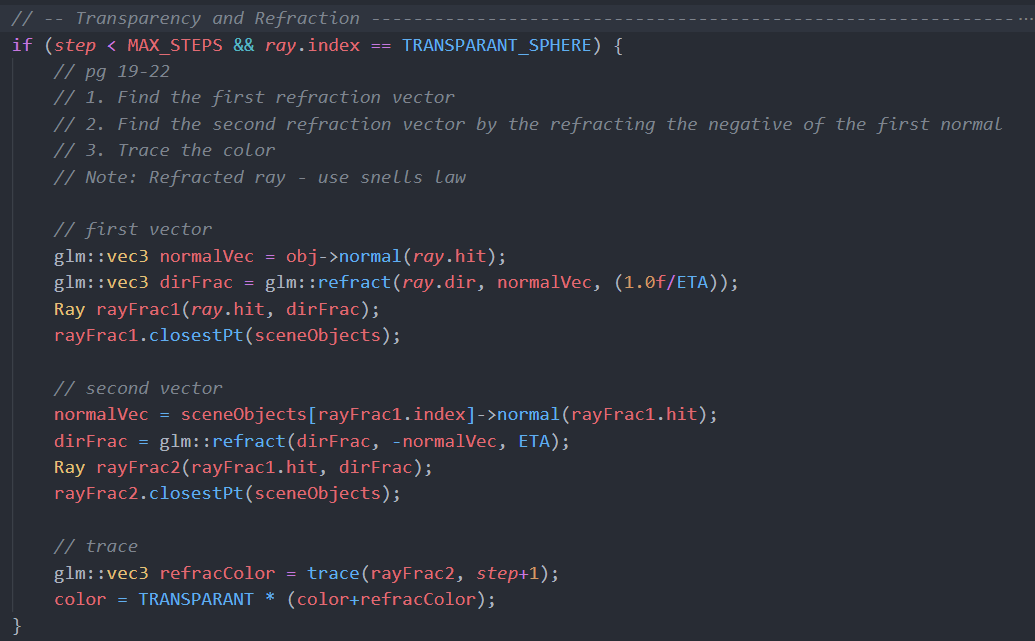
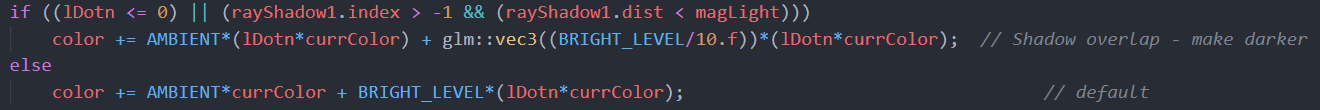
**

Figure 5 – Taken from lecture 7, slide 5

1. **Light Sources – shadows**

Figure 4.2 – Code for part 4

* There are two light sources in the scene.
* Light source 1 reflections are made up of *ambient + diffuse* (shown in fig 5 and 6) and a brightness of 30 %
* Light source 2 is made up of *ambient + diffuse + specular* but reflections are just *ambient.* At 100 %

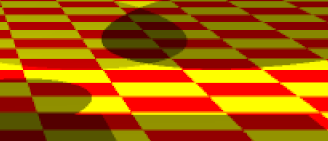


Figure 6 – First two lines controls shaded area – shown below

1. **Non-Planar object textured**

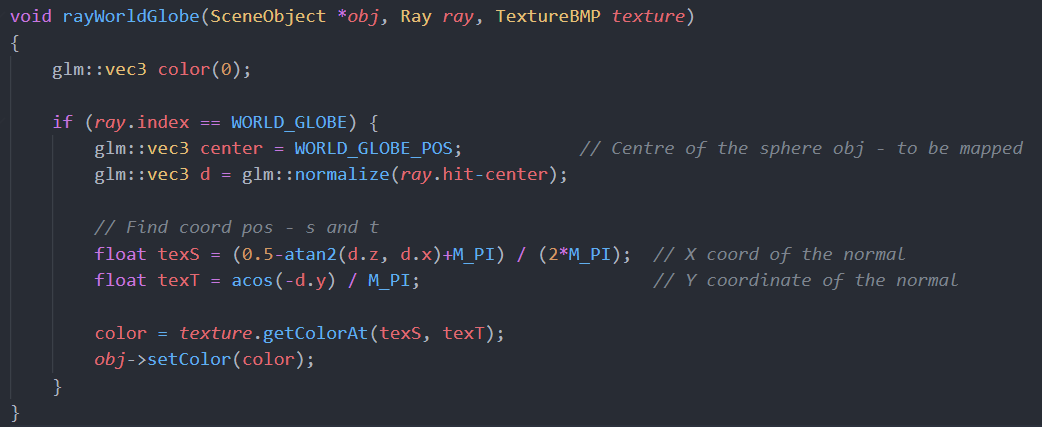
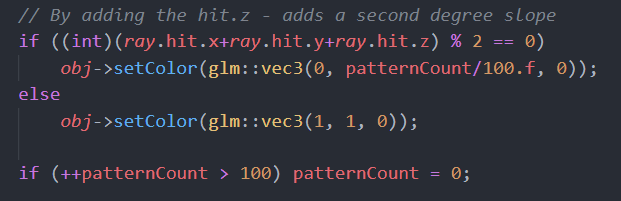
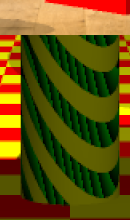
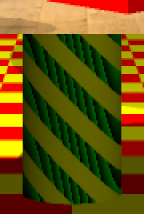
Earth.bmp is mapped to the sphere – algorithm in function – *texS* and *texT* lines

Figure 7 – Shadows overlapping



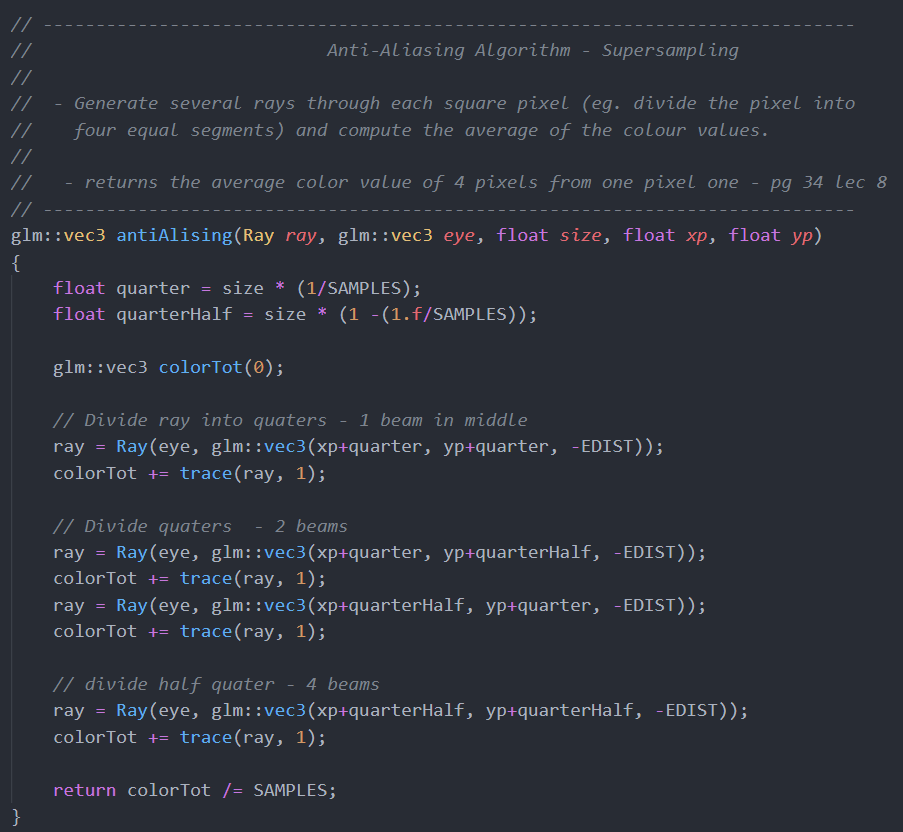
1. **Non-Planar object textured pattern**

The Cylinder is textured by the code above

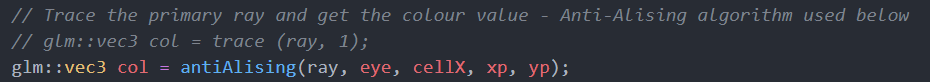
2nd degree slope- quadratic

- Note: patternCount is random and changes the color pattern when

the window is clicked on or moved – this is because ray is refreshed.

****

1. **Anti-Aliasing – Super Sampling**

Algorithm:

Code and Usability

1. Square Pixel is divided in to 4 sub pixels – 1 beam.
2. Two beams are generated by dividing again.
3. 4 beams generated by dividing by a quarter and a half i.e 0.75

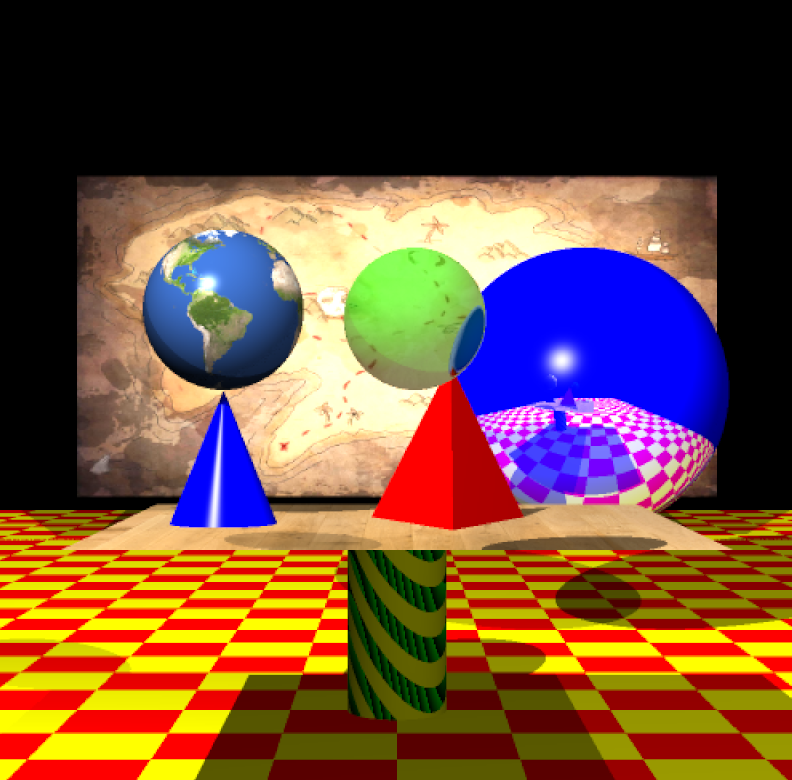
Figure 8 shows Aliasing enabled and 9 is with no aliasing

Fig 8 - Aliasing enabled

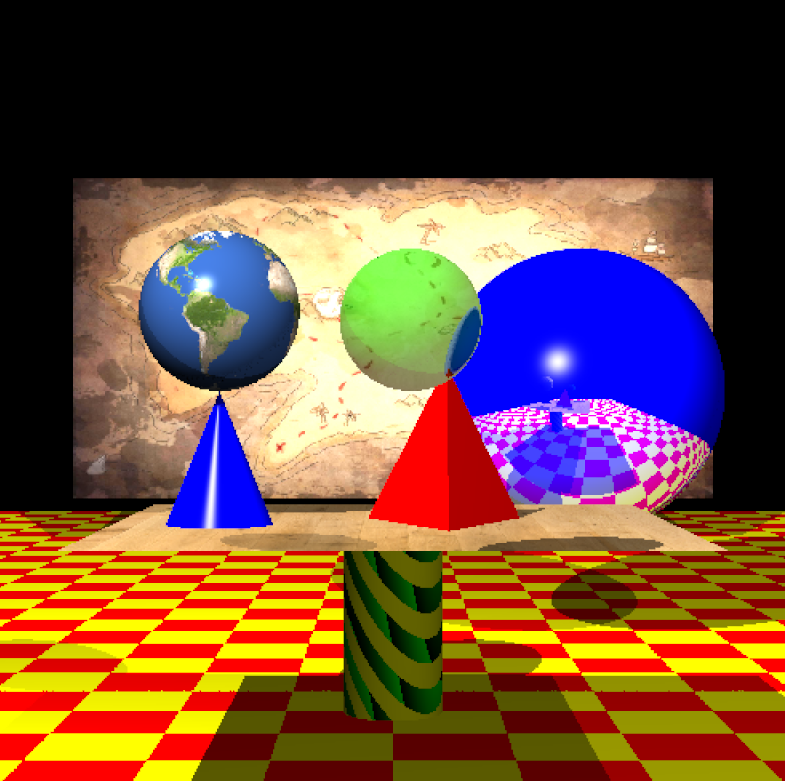


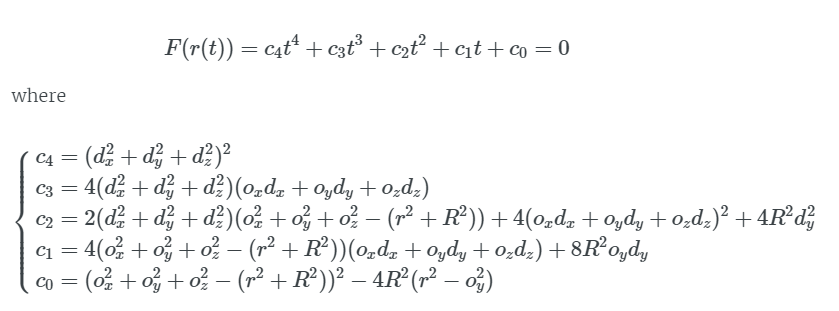
Fig 9 - Aliasing disabled

1. **Success And Failures**

The Scene was planned to have more objects at the start but decided to cut back and focus on the quality of the main parts of each object in the scene rather than quantity.

I was able to create all my objects in a *SceneShapes.cpp* which allowed me to easily go back and find parts of coded that needed fixing.

I was very happy with my:

* Shadows
* Earth sphere map and other textures
* Cylinder, Cone and planes
* Refracted and transparent object
* Pattern generation
* Simple effective Anti-Aliasing Algorithm
* Good scene design

<https://marcin-chwedczuk.github.io/ray-tracing-torus>

Challenges – tried:

* I tried to create a torus but found it was very complex and had a lot a issues with the intersect method. At the end, I had to move on but it had just to many roots for me to solve and code

texture[0] = TextureBMP("../Models/Earth.bmp");*// http://www.world-maps.org/*

texture[1] = TextureBMP("../Models/Table.bmp");

*// https://freestocktextures.com/texture/floor-wood-oak,765.html*

texture[2] = TextureBMP("../Models/Treasure\_Map.bmp");

*// http://www.aljanh.net/map-pirate-wallpapers/1436032319.html*

Images reference