Assignment 3/4

<u>Process to complete the assignment</u>

Our process to complete the assignment was the following and the work was split up / worked as indicated in the 'Roles' section:

- Understand the starting code while referring to lecture notes / supplementary material
- Complete the ray casting / first hit code first
- Compute the intersections for rays and spheres
- Implement the illumination models (phong illumination, shadows, reflections)
- Complete the requirements for Assignment 3 as early as possible
- Add in the assignment 4 functions.
- Draw a more complicated scene.
    - Obj files were downloaded from free 3d object sites
    - The objects were edited as desired in Maya
    - The scenes were constructed in Maya to determine the appropriate transformations of the objects

<u>Description of the code and how it works</u>

Our code is object oriented.

In the main folder, you will find:

- Main.cpp
    - Builds the scene
    - Initializes the camera coordinates
    - Initialize the skybox
    - Instantiates the ray tracer
- RayTracer.h/.cpp
    - Does ray tracing
    - Handles all object traversals and intersection shading
- In ObjectTypes folder, find
    - Object.cpp
        - Provides base class for all other objects to inherit
        - Has function prototype for finding closest intersection to a given ray
        - Holds general information that every object has, such as material, colour and texture.
    - Intersection.cpp
        - The object that is returned when an intersection is found.
        - Contains all relevant information such as point of intersection, normal, colour, etc
- In ObjectTypes/ObjectSubclasses folder find:
    - All implemented object subclasses along with their helper classes. Of particular note there are:
        - TriangleMesh.cpp
            - Provides support for arbitrary triangle meshes.
            - Can even interpret .obj files with multiple meshes, materials, and textures (i.e. if the .obj file specifies multiple sections for an object each with different materials such as a wine bottle containing wine, the TriangleMesh class will still be able to display the object as intended).
        - BoundingBox.cpp
            - The bounding box volume hierarchy applied to a mesh that GREATLY accelerates its rendering time
        - OBJ_Loader.cpp

        - Utility code to parse an .obj file into a suitable mesh structure, obtained from https://github.com/Bly7/OBJ-Loader
- In the GeometricTypes folder, you will find:
    - geometric types such as homogenous point and affine matrix classes.
- In the Lights folder, you will find:
    - the two types of lights implemented, point and area.

Easy vs Challenging

The hardest part of the assignment was handling arbitrary mesh geometry. This aspect of the assignment was challenging as obj and mlt files have varying structures. Obj may store several meshes and/or provide texture coordinates for several different textures. Storing and accessing this information efficiently and effectively. Another challenging part of this assignment was runtime. In order to keep the time required to render a scene low, the code was required to be optimized (for example, include bounding volumes around the triangle meshes).
The easiest part of this assignment was implementing the algorithms taught in lecture. Incorporating the Phong model, the intersections with the sphere and plane, were proven to be some of the easiest parts of the code to add as the pseudo code was provided.

Course Understanding
This assignment helped us further understand geometry and how to compute intersections and normal for various shapes. The assignment also provided a more intuitive understanding of the importance of switching between camera and world/local coordinates.

Role
Grace
- Intersection for ray-plane
- Phong illumination
- Implement a neat scene
- Report
- Handing arbitrary surface mesh geometry

Eric
- Reframed the starting code
- Intersection for ray-sphere
- Shadows/Reflections
- Anti-aliasing
- Area light sources
- Multi-thread
- Refraction
- Glossy
- Texture-mapping
- Adding environment mapping

Checklist A4:

Anti-aliasing	Complete
Area light source	Complete
Neat Scene	Complete
Report	Complete
Advanced techniques	
Handling arbitrary surface mesh geometry	Complete
Glossy Reflections	Complete

Texture mapping	Complete
Adding environment mapping	Complete
Multi threading	Complete
Refraction	Complete
Bounding Volume Hierarchy	Complete