3GC3 Fall 2023 - Assignment 4
RayTracing - Diffuse and Specular Spheres

In assignment 4, we will implement some key functions in the ray tracing algorithm. You are given a folder of source code for this assignment, with some “TODO:” marked in the code. Fill in the missing part in the “TODO” section, and make the ray tracing algorithm work.

1. Project Overview

In assignment 4, you are given one **main.cpp** file together with **Vector3D.h**, **Camera.h**, **Ray.h**, **World.h**, **Sphere.h** and **Material.h** header files. Please create an empty C++ project using the provided main.cpp and include all the header files. Build your project and make sure that there is no error at the beginning status, otherwise adjust the code accordingly.

The provided code has the majority of the ray tracing algorithm implemented, with some missing pieces. Your job for assignment 4 is to implement the functions that are marked with “TODO:”.

Here is some brief explanation of the code: starting from the main function, an image of size 768x540 is defined for ray tracing. A camera is looking at the world, emitting rays through each pixel in the image. The world is composed of a list of spheres, made of diffuse or specular materials. For the ray tracing algorithm to work, each ray has to test its intersection with each sphere to compute image pixel color.

2. Implement the TODOs

Each “TODO:” has an id associated with it, make sure you follow the order.

“TODO: 1” - Set Result Image Path

Find a local folder on your machine, and specify the output ppm image path in the main function in main.cpp

“TODO: 2” - Sphere Hit

In Sphere.h, please fill in the implementation for Sphere::hit. This function computes if the ray hits the sphere within the range min_t and max_t. Fill in the information in HitResult hit_result. If there is no hit, assign hit_result.m_isHit = false, otherwise hit_result.m_isHit = true and compute the detailed hit information such as t, position, normal, material, and etc.

“TODO: 3” - World Hit

In World.h, please fill in the implementation for World::hit. This function computes how the ray hits the world within the range min_t and max_t. As the world is composed of a list of spheres, so in this World::hit, you will need to iterate over each sphere in the world, and call the sphere->hit(...). Note that only the nearest hit information is needed.

“TODO: 4” - Diffuse Material
ReflectResult reflect(Ray& ray, HitResult& hit) computes how the ray reflected on the material at the hit point. “TODO: 4” is for the Diffuse material. As diffuse material has equal probability to scatter the incoming ray in all different directions, so in this reflect function, you need to generate a scattered ray that starts from the hit point and shoots toward any random direction on the hemisphere.

“TODO: 5” - Specular Material
For the Specular material, the incoming ray will be mirrored around the normal of the hit point. So in this reflect function, you need to generate a mirrored ray that starts from the hit point and shoots toward the mirrored direction.

“TODO: 6” - Generate Results
In the main function in main.cpp, there are several predefined worlds:

World	Save rendered PPM to name
world.generate_scene_one_diffuse();	1diffuse.ppm
//world.generate_scene_one_specular();	1specular.ppm
//world.generate_scene_multi_diffuse();	mdiffuse.ppm
//world.generate_scene_multi_specular();	mspecular.ppm
//world.generate_scene_all();	all.ppm

Uncomment the code, render the worlds one by one and save the rendered ppm images to the names listed in the above table.

Submission Checklist

Submit your **code** together with all the captured ppm images.

1diffuse.ppm - 5 points
1specular.ppm - 5 points
mdiffuse.ppm - 10 points
mspecular.ppm - 10 points
all.ppm - 15 points

code readability - 5 points
TODO 2 - 20 points
TODO 3 - 10 points
TODO 4 - 10 points
TODO 5 - 10 points
Submission deadline is Dec. 14th, 2023