Assignment Report #7

Time Spent: 15 Hours

For my final assignment, I decided to update my raytracer by using antialiasing, diffuse materials, implementing metal as a material, dielectrics, motion blur, textures and several other features. These features helped me to create a much more robust raytracer, however at a cost with performance as I am rendering a lot of visuals that will take far longer than need be. I added a way of tracking progression for the raytracer and it does render, but the amount of time it takes can be a bit tedious and time consuming. I advise not rendering too many ppm files in order to not consume too much time. I believe another reasoning for the raytracing taking far too long is the aspect ratio being 1:1 with a pretty broad field of view, 800 by 800, making for a lot of shapes and area to render. I tried to implement a way of keeping the runtime very low by creating a samples per pixel variable that would take quality down a bit, but the runtime still can reach very high times, pending on the samples per pixel I once had it reach 15 minutes which was very difficult to consistently debug and test.

For the rays to help simulate the behavior of light in a virtual 3D environment, I used a parametric equation for a ray. I would get a point in 3D space along the ray by having the origin of the ray added to the value of a direction vector of the ray by a value that would control how far along the ray was. This would allow me to determine what color is seen along a particular ray.

For the diffuse material, another simple method was implemented simulating diffuse reflection. For this method, a ray that hits a diffuse surface has an equal probability of bouncing in any direction away from the surface. The randomness in the direction of reflection is what gives the material its matte look. This method is intuitive, but not the most accurate. Also, there was the need to fix shadow acne.

With the issue of aliasing, I use antialiasing by involving a sampling of a square region around each pixel, extending halfway to neighboring pixels. This would allow an approach that would capture a more accurate representation of the continuous light falling on the pixel and is acknowledged as a simple yet straightforward method. This is however not optimal, as more sophisticated anti-aliasing techniques exist. There could be problems capturing complex patterns and intricate details in the scene.

For the Metal, an abstract material class with unique behavior was implemented to allow for material types to be encapsulated in their own class. This would allow for inheritance and define essential behaviors for materials to exhibit.

With using dielectrics, Snell’s law was used for determining the direction of a refracted ray by governing the refraction of light at a boundary between two different media with different refractive indices.

For motion blur, I am sacrificing runtime speeds to make up for the visual quality. This is due to brute forcing the effects to some degree. The use of multiple samples per pixel for rendering the effects is implemented. This is done by casting multiple rays for each pixel to capture a more accurate representation of these effects. As a result, the white sphere simulates movement as if the sphere is in motion while the shutter to a camera is open.

For the textures on the sphere to simulate a globe, I applied a jpg to a sphere by taking the coordinates of the sphere using a stb\_image utility to read in image data in an array of unsigned chars. This image that is read in is then assigned to a lambertian material and applies it to the sphere to give us a picture of the earth.

Despite trying to improve my raytracer, the time to render is still very time consuming and I highly advise not converting to a .ppm file multiple times. I implemented a make file to allow for compilation to be effortless. To get a .ppm file, simply type ‘./raytracer > image.ppm’ into the console and the rendering process will begin.

References:

* https://raytracing.github.io/books/RayTracingInOneWeekend.html
* https://www.youtube.com/playlist?list=PLlw1FcLpWd41aG18PMtlakvT3MQEQizkn

