Amine Roudani

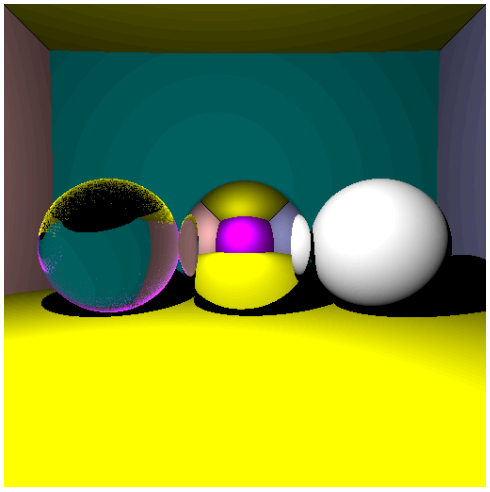
CSE306 - Computer Graphics

Assignment 1

TD 1

In this lab, we learned how to render spheres. I had never done anything like this before which meant that this lab was super insightful and interesting. For a brief overview, the main classes include `Vector` for handling geometric computations, `Ray` for simulating light paths, and `Sphere` for representing objects in the scene. Each `Sphere` can be defined with specific optical properties like being a mirror or having diffuse reflection. The `Scene` class aggregates these objects and handles the rendering logic, determining color based on light interactions. The rendered scene comprises various spheres with different properties positioned strategically to demonstrate reflections, shadows, and basic lighting effects, providing a foundational understanding of ray tracing concepts.

Our image for this lab is as follows:

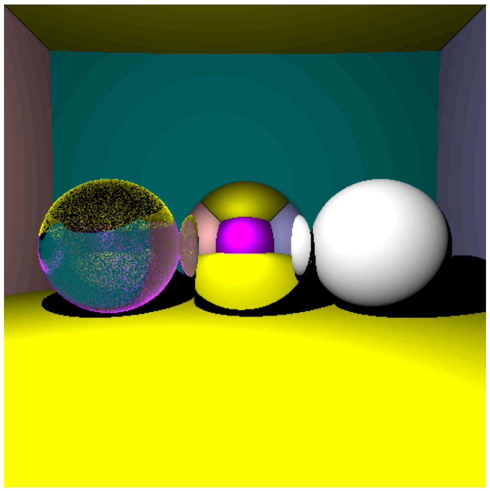


Picture 1

Left: Sphere with Refraction, Middle: Sphere With Reflection, Right: Sphere

Time: 9045 ms

After this, we added Fresnel Reflection according to the Fresnel law in the lecture notes. The following image includes Fresnel Reflection.



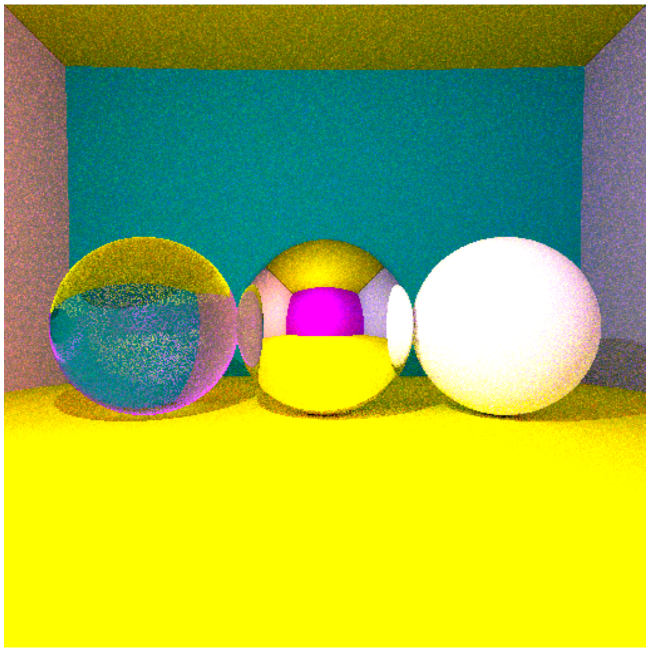
Picture 2

Adding Fresnel Reflection

Rendering Time: 9899 ms.

TD 2

In this TD, the first goal was to implement Indirect lighting for point light sources without Russian roulette. This was done by creating a new function *generateRandomDirection* part of the Vector class and updating the color in getColor using this new function. The following image shows the addition of indirect lighting.

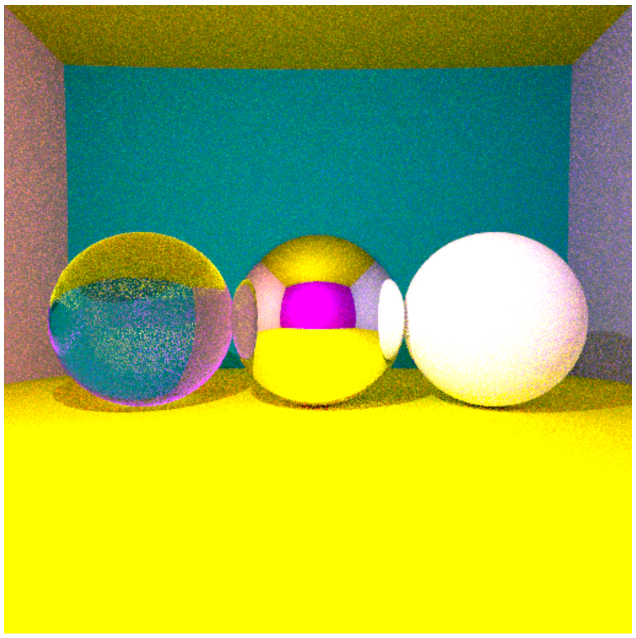


Picture 3

Adding Indirect Lighting

Rendering Time: 53341 ms

The second part of TD2 involved implementing antialiasing to make the image look 'smoother'. Antialiasing was implemented using supersampling where multiple rays are cast per pixel from slightly different starting positions within the pixel to capture more scene detail and average out the final pixel color. This reduces the visual artifacts of aliasing like jagged edges. The image below shows the results



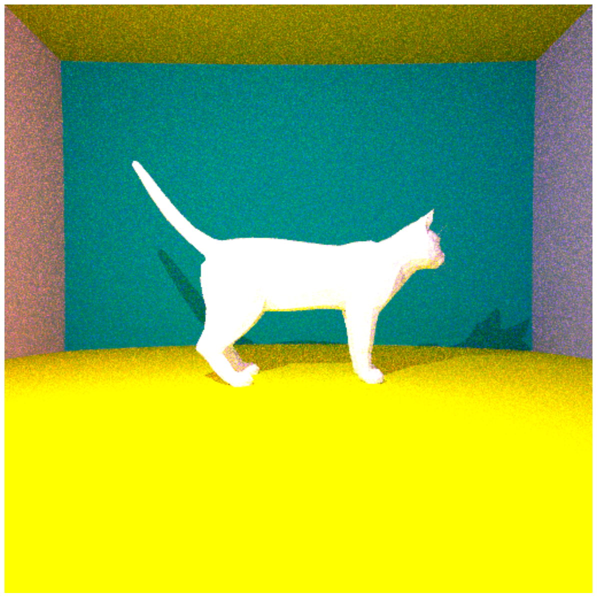
Picture 4

Adding Antialiasing

Rendering Time: 53066 ms

TD3

During this TD, we moved onto implementing TriangleMesh and TriangleIndices. I used the code in the lecture to do this. I also implemented a new parent class Geometry that would go above Sphere and TriangleMesh. Amongst other things, I also had to make sure that Ray intersections were handled. Below is the cat from this TD.



Picture 5

cat.obj

10013291 ms (approx 16 mins)

TD4

This was the final lab for this project. It was all about implementing BVH to the ray-mesh functions in order to save considerable computational time. Below is the final result.

Picture 6

Same cat as before, now using BVH

Time: 313029 ms (approx 5 mins)