Raytracer (2021) In this report I am going to be explaining how I achieve version 1.0 of my raytracer, which includes Blinn-Phong materials, shadows, two lights, reflection and refraction.

First, shadow implementation. The shadow is made by the creation of a ray the calculations of the ray depend on the type of light. Then crate an intersection using that ray and if at the end of the rest of the calculations there is and intersection, the pixel color is black.

Then to calculate Binn-phong we need to calculate specular with a half vector: H = L + V / magnitude of (L + V), then (N dot H)ŝhininess where L = light, V = viewerm N = normal [1] that value gets added along with ambient and defuse to get Binn-phong

To calculate reflection I created a method whe i used the formula: R = I - 2N(N dot I) where I = incident ray, N = normal, then calculated the coalitions with than new intersection. This method is recursive so the reflections include other reflections

I did another method for refraction where I used T = nI + (nc1 - c2)N where n = 1/ refraction index, I = incident ray, c1 = N dot I, c2 = $\sqrt{1 - n * (1 - c2^2)}$ [1] then calculated coalitions with the new intersection, I also had to make this method recursive, and include it in my reflection method do the refractions could be reflected as well, also implemented the reflection method in this one.

The rendering process look a lot of time for complex objects and bigger resolution. I had to be extra careful because if I didn't like it i had to run it again and wait a few hours.

With that i concluded version 1.0. It was very challenging, and I experienced a lot of issues when i was developing my code, but at the end I feel really proud of the outcome and I think that this project made me learn a lot and believe in myself a little bit more. I really liked it even though sometimes I wanted to cry about it. I am excited to put in practice what I have learned and show everyone my renders.

References

[1] ScratchPixel. Introduction to shading.