

Computer Graphic - Ray Tracing

Bonus Implementations:

- Sphere

A ray can intersect with a sphere with 0 intersection points, with 1 intersection point (if the ray tangent the sphere) or with 2 intersection points the first one is when the ray came from the outside of the sphere and the second from inside the sphere.

We use the implicit formula of sphere and place the ray source point (p) and direction(tv) to get a quadratic equation to find the two solutions for t. Then we select the minimal t (to get the closest intersection point).

- Refraction Surface

We checked if a surface is transparent, if it is, we calculate the ray that breaks from this surface using snell's law. We implement the refract method in the Ops class, we find the cosine(theta2) using a formula from [wikipedia.org/wiki/Snell](https://en.wikipedia.org/wiki/Snell's_law). Then in the Scene class we get all the parameters of the surface (the normal, n1,n2) and use the refract function to get the T ray (the break ray that is discarded from the hit surface).

- Super Sampling

To implement the super sampling we added a check that checking if the antialiasing parameter is larger then 1, if it did we created more (antialiasing - 1) new rays in a random place in the pixel. We pass the pixel width parameter from the PinholeCamera class and randomly choose a step in half the size of the pixelWidth up or down(-up) and right or left(-right).

After we created each new ray we calculate its color and add it to the origin color (from the center ray we already calculated). At the end we distributed the color by the number of rays we shooted through one pixel (which is exactly the antialiasing parameter).