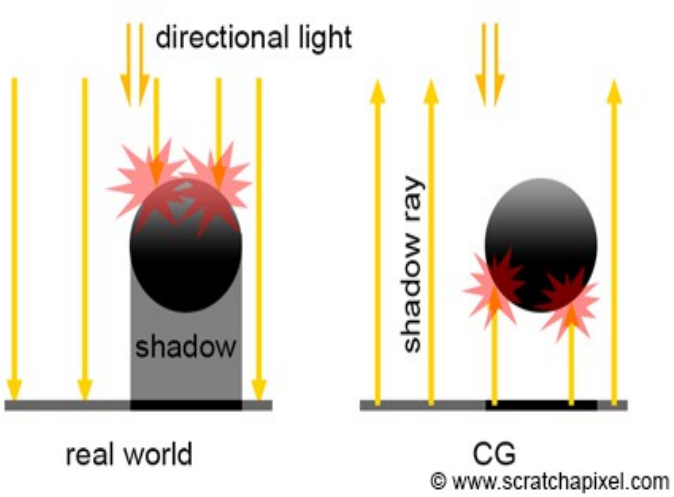
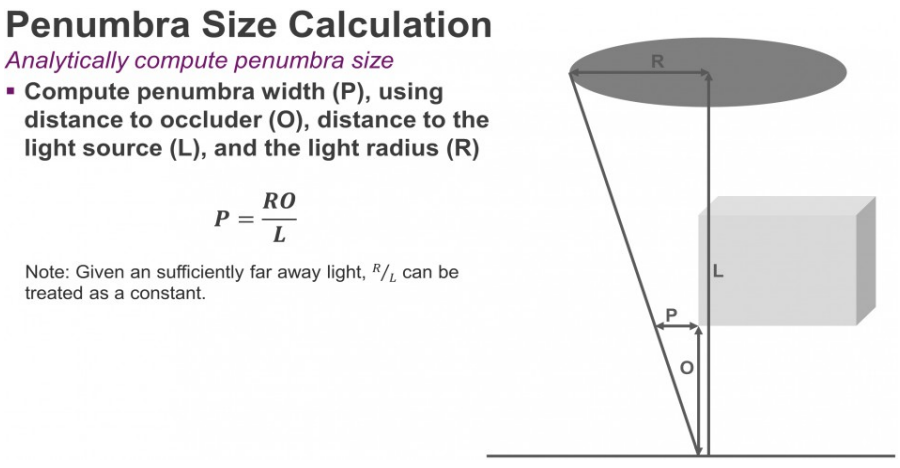
For the plus one I attempted to implement a soft shader. In real life a light is blocked by an object and that creates a shadow on an object. To implement this in a computer enviroment however it is a little different. You have to determine if a specific point is in the shadow of another object. As demonstrated in the picture from scratchpixel below:



To do this, I started the ray tracer from the camera point of view and shot it towards each individual pixel and returned the time the ray intersected . I then used that time in the equation of the ray in order to find exactly where the ray intersected the object in 3d space I then shot a ray from that point to a light source. If that ray intersected an object I than determined that it would be in the shadow of that object and darkened that pixel.

If I could of gotten this to work I would of liked to implement some methods for softening the shadows. The first, simplest thing I would of liked to implement is to take into consideration the distance of the object that is being shadowed and scale the shadows based on that distance.

The really cool thing I would of liked to implement would be using the penumbra and umbra described by imgtec.



This would result in more realistic gradient shadow.

Sources:
https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-shading/ligth-and-shadows
https://www.imgtec.com/blog/implementing-fast-ray-traced-soft-shadows-in-a-game-engine/