Assignment 2

OS: Windows 10 (64-bit)

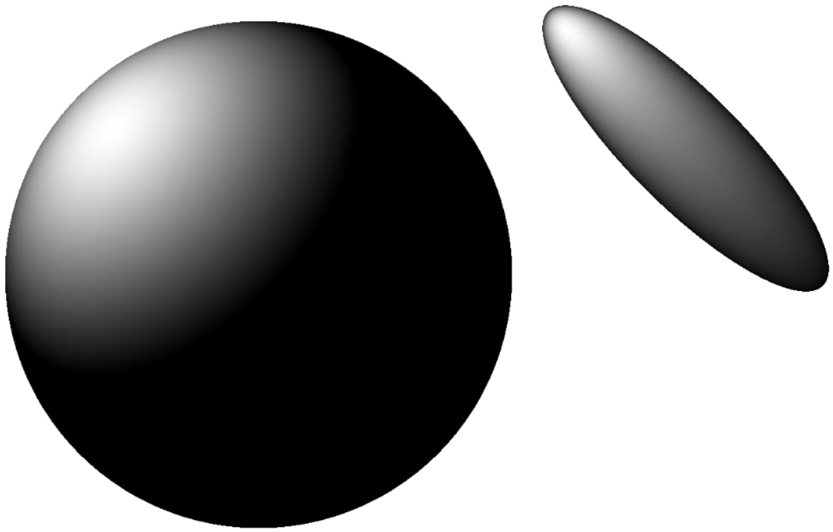
Compiler: Visual Studio Code

Ex.1: Basic Ray Tracing

1. Parallelogram created on z’s 0.5 plane in lines 151-154 of main.cpp.
2. Created a function intersection in lines 132-137, which solves for parameters needed to find an intersection of a parallelogram with a ray based on the formula given during the lecture 4.
3. In line 169 if the parameters found using above function satisfy the conditions discussed during the lecture 4, then it means the ray intersects the object and we can use one of the parameters and the equation of the ray to find the intersection point (line 171).
4. In line 174 we find the normal just by calculating the cross-product of parallelogram’s sides.
5. For perspective projection, we set ray’s origin to the position of the camera in line 216. A ray’s direction is found by subtracting a camera’s origin from pixel’s origin. NOTICE: since a camera is pointing in the direction -z, I’ve projected all the objects to the plane z=0.9. So the picture is actually a square [-1,1]x[1,-1] on z=0.9. The shape of the sphere would’ve not changed if the camera was directed towards its center.
6. For the parallelogram:

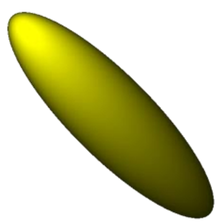
Since the projection space was much closer to the camera than the parallelogram for perspective camera it became much smaller.

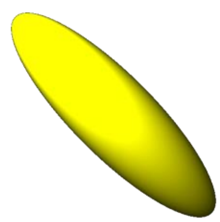
For the sphere also since the projection space was closer to the camera than the parallelogram for perspective camera it became smaller. Also, we can see that since perspective changed the light on the sphere became larger. As I’ve already mentioned earlier since I was projecting the sphere to the plane on z, it has changed its shape.



Ex.2: Shading

1. Ambient is constant, which is written in line 259. k_d in line 283 is a coefficient for diffuse and k_s in line 284 is a coefficient for specular. In line 286 diffuse is implemented. In line 287 specular is implemented. For h we've just summed the vector pointing to the camera from the intersection and the vector pointing to the light from the intersection (in our case it's the same vector).
2. RGB component added in line 302, which made a sphere yellow.
3. Experiments with different coefficients:





First picture corresponds to k_d=k_s=0.1. Second one corresponds to k_d=k_s=0.1. Third one corresponds to k_d=2, k_s=0.5. As we've expected when we increase k_d and k_s, shadow on the sphere decreases since the intensity of light increases.