

CSE306 Computer Graphics

Xinyi DAI

Code:

-main.cpp: The main function

-Vector.h, Sphere.h, Intersection.h, Image.h, Ray.h : The defined frame class

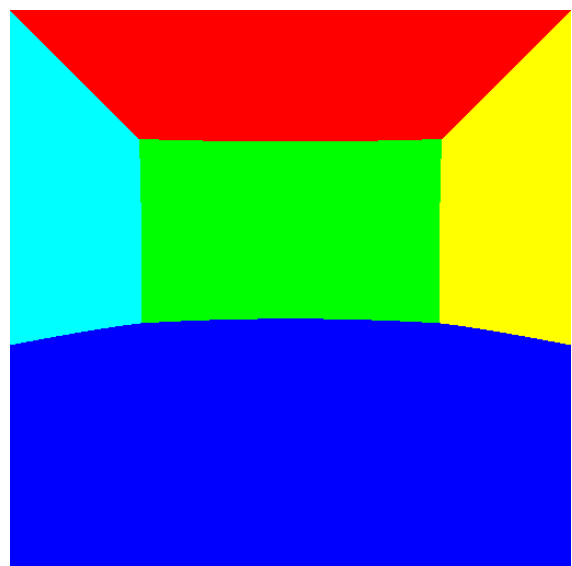
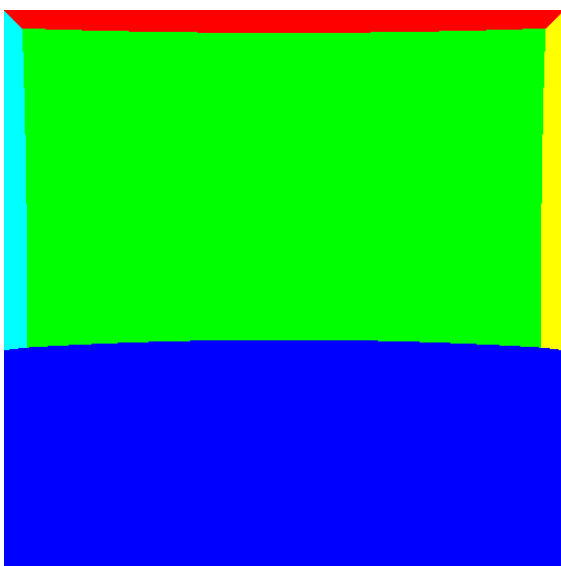
-Scene.h: the Scene is the main part to realize the graphics calculate.

1. Preparation:

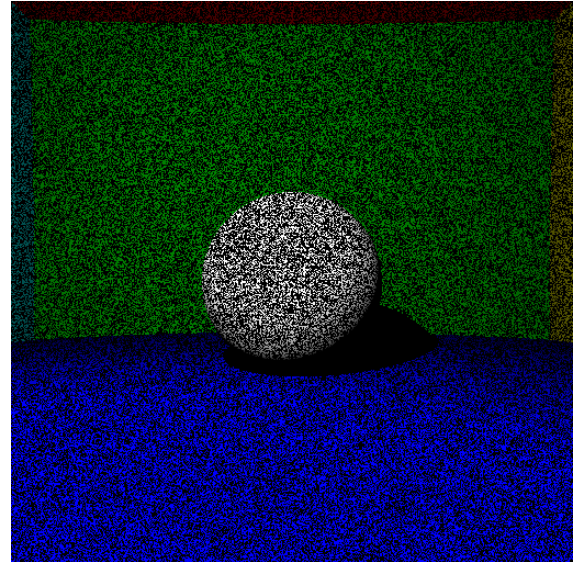
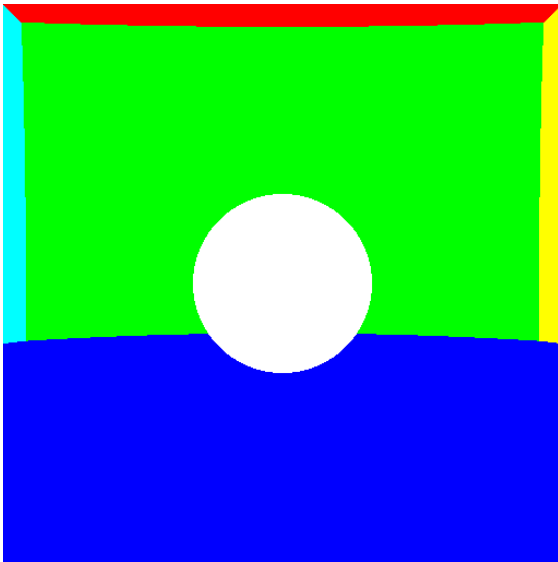
Write the Structures especially the vector class. Vector class is very simple but extremely important, it's the basis of everything. From my personal experience, many bugs caused by some small error in vector cost me hours to debug.

2. Raytracing / Path-Tracing

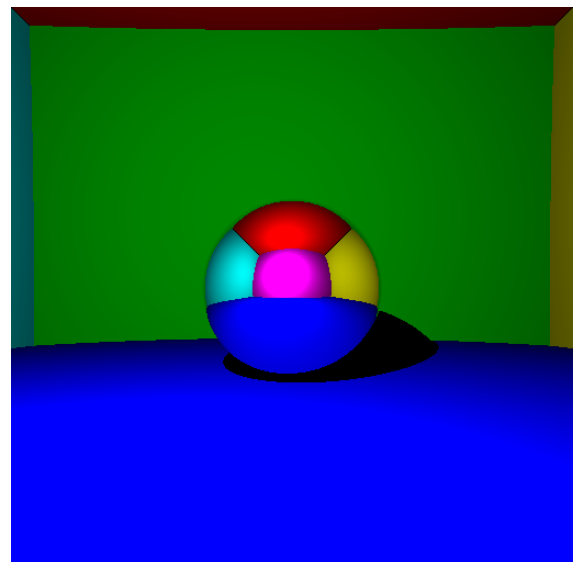
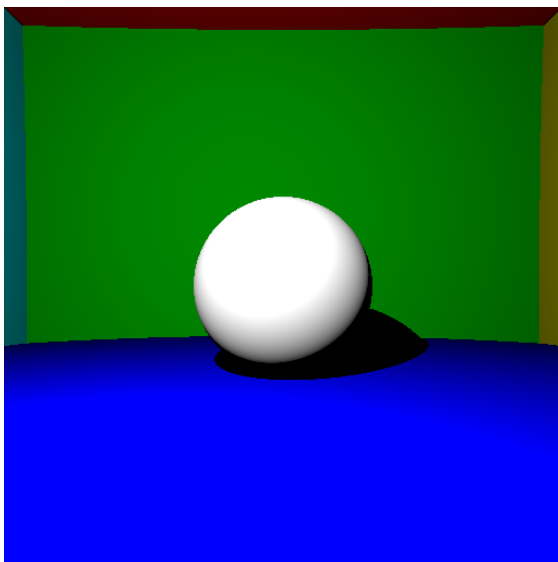
First we define the scene class, create a initial scene with only walls using the parameters: $H=W=512$, $fov=\pi/3$. We find that if we want a bigger horizon, we can use a bigger for ex. $fov=\pi/2$ in right picture. Then write the intersect of ray for sphere. Check the correction by directly return the color of the point of intersection in the function getColor



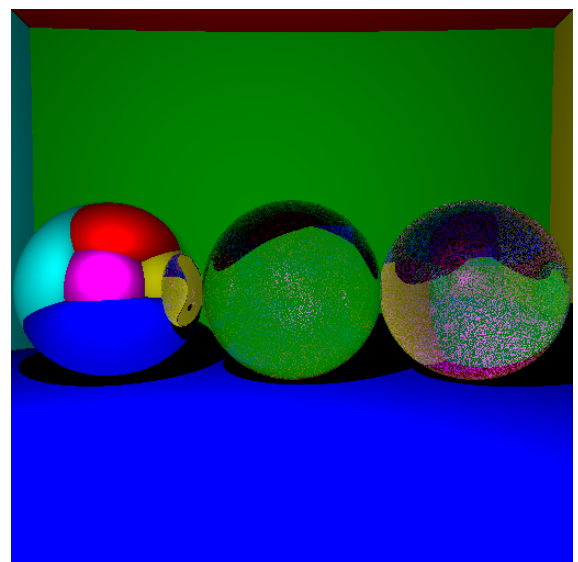
Add a white ball in the centre(left) then write the getColor for diffusion:



We reduce the noise by adding small epsilon, however we should take care of the direction of norm. We introduce a parameter in sphere to define its characteristic: -1 for diffusion(left), 0 for mirror,(right) $n > 0$ for the transparent sphere with refractive index of n .



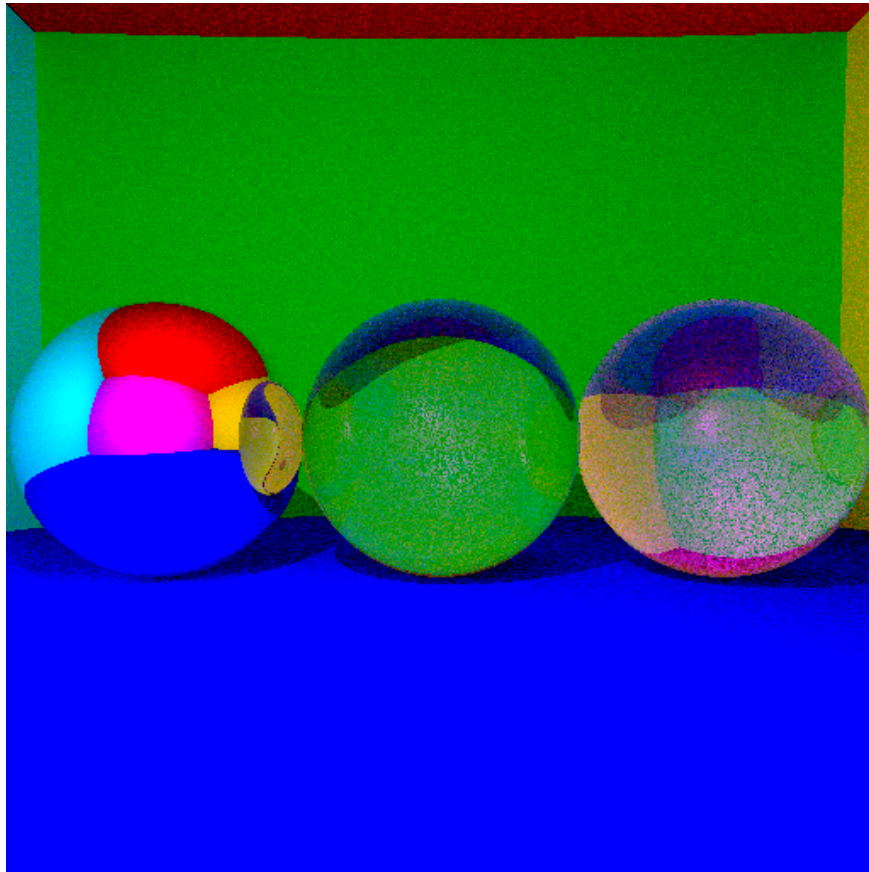
In the refraction part, we should notice that in certain time the ray is inside the sphere. the following picture of three balls are mirror(left), transparent with $n=1.3$, transparent with $n=1.7$. For every pixel, because of low cpu, i only run for 10 times, therefore the picture is quite jagged.



3. Adding indirect lighting

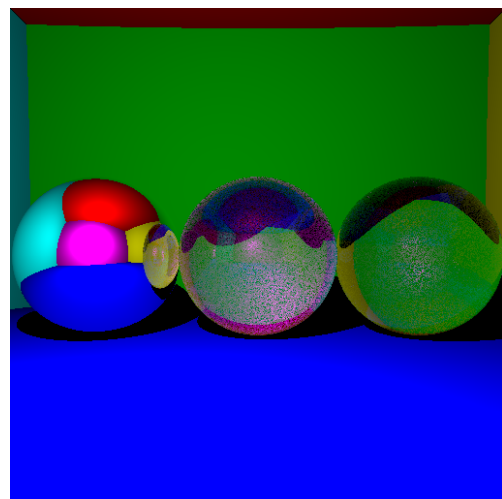
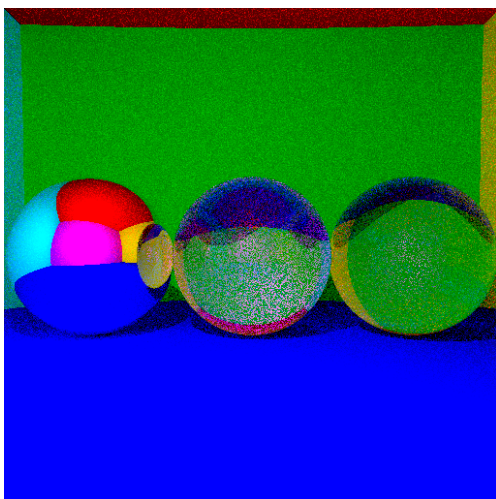
Now, we add the indirect lighting for point light sources.

The following results were using parameter `ray_number=35`, `ray_depth = 5`:



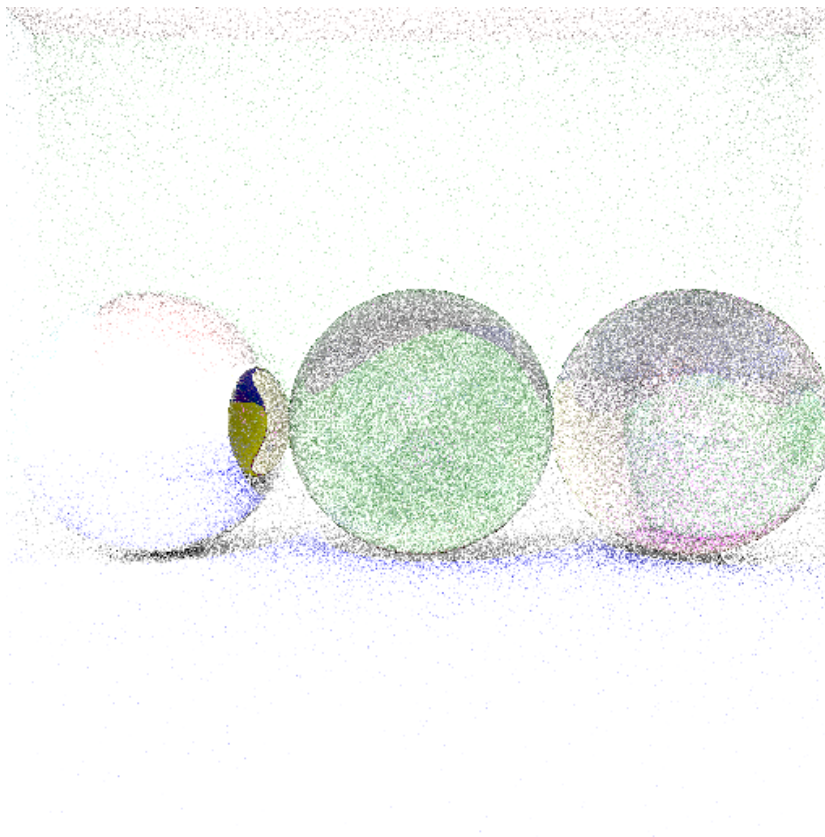
4. Antialiasing

we test on the tribal model with no indirect-light and middle ball $n=1.7$, right ball $n=1.3$
number of ray =10(left) and 35(right)



Some Art work by bugs:

When I don't define the * for the `albedo` * `getColor` in indirect light but use dot



Define * by return vector($a_0 \cdot b_0$, $a_1 \cdot b_0$, $a_2 \cdot b_2$):

