Ray Tracing Project

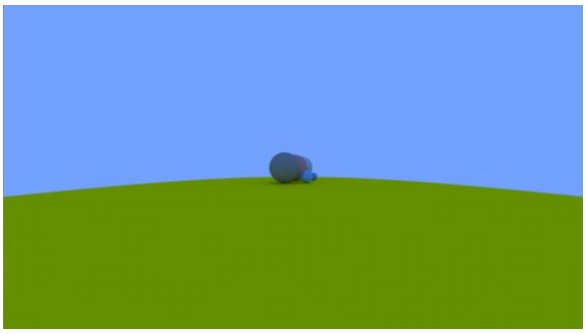
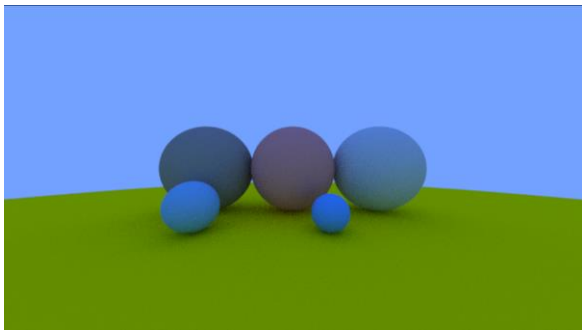
Baran Çimen
26336

Task 1 [Required]: Basic Scene (15Pt)

I have created 6 spheres to create a basic scene. All has the same material, Lambertian. Their positions and radiuses are:

```
(vec3(0.0, -100.5, -1.0), 100.0,
(vec3(0.0, 0.0, -1.0), 0.5, mate
(vec3(-1.0, 0.0, -1.0), 0.5, mat
(vec3(1.0, 0.0, -1.0), 0.5, mate
(vec3(0.3, -0.35, -0.3), 0.15, m
(vec3(-0.8, -0.3, -0.3), 0.2, ma
```

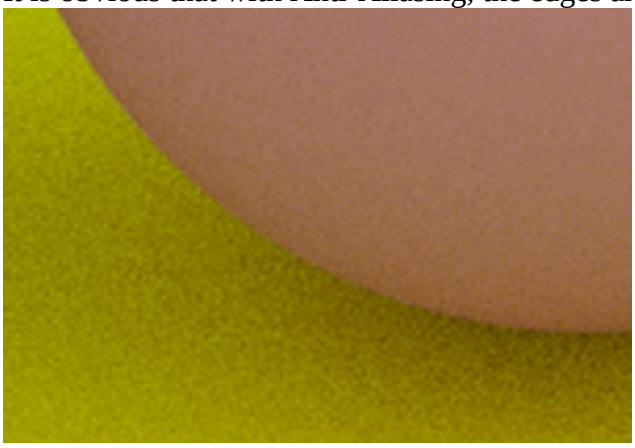
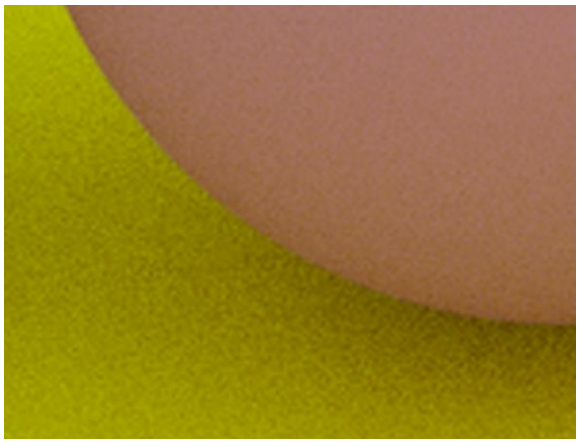
The camera gets closer in x axis in the second picture

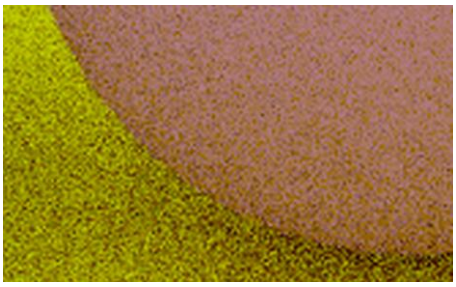
Task 2 [Optional]: Anti-Aliasing (10Pt)

I have used random algorithm and grid algorithm for supersampling. On the left is random algorithm's view, and on the right there is grid algorithm's view. I have used sample size of 100 for renders. Beneath them, there is the render which has sample size of 5.

It is obvious that with Anti-Aliasing, the edges are more smooth.

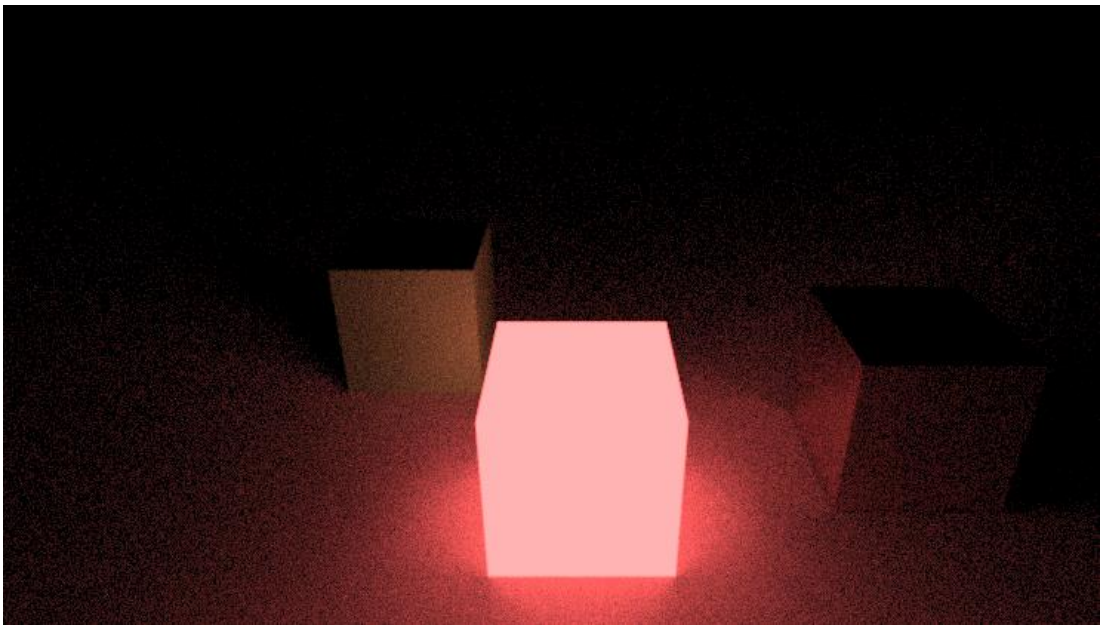
Random Algorithm *Grid Algorithm*



Sample = 5 Render

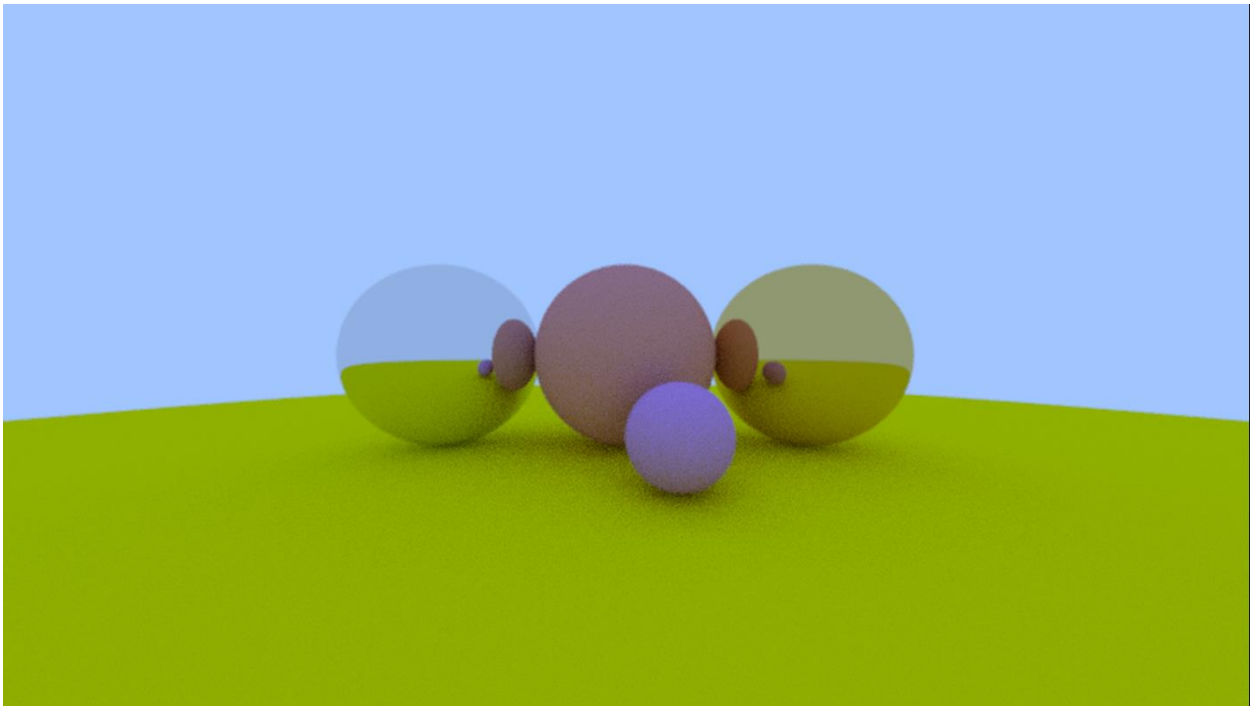
Task 3 [Optional]: More Shapes (5Pt)

I have created box class with xy-yz-xz planes. The one on the left is metal while the one on the right is Lambertian. The middle box is the only light source.



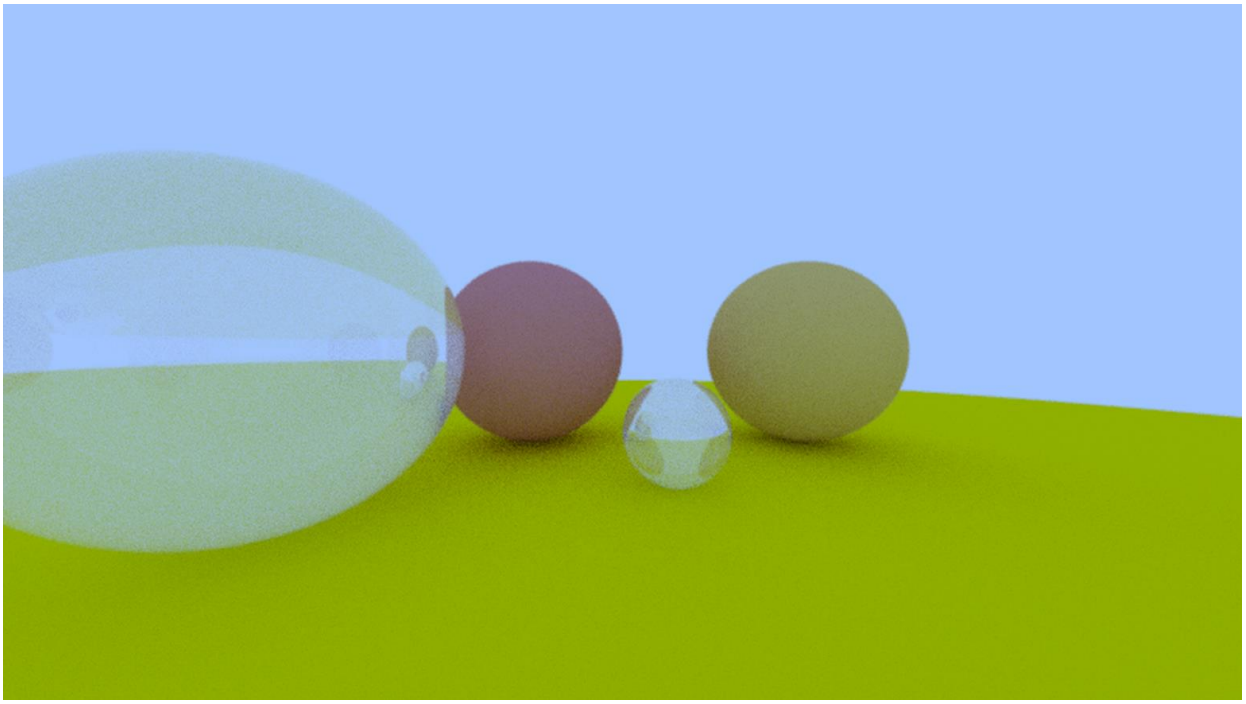
Task 4 [Required]: Diffuse and Metal Materials (25Pt)

I have created 2 metal and 3 Lambertian spheres. One of the Lambertian spheres is the ground where the other spheres stand on. The left and right spheres are metals, and they have different color values. Metal scatters the light with respect to the normal of the intersection while Lambertian scatters with some randomness.



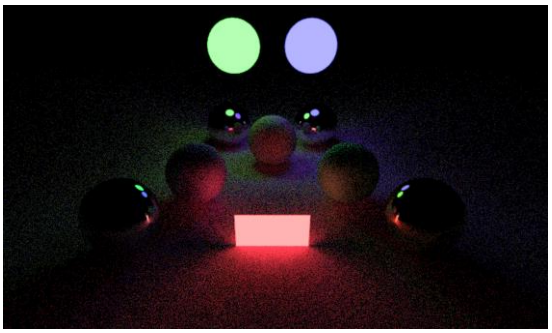
Task 5 [Optional]: Refraction (10Pt)

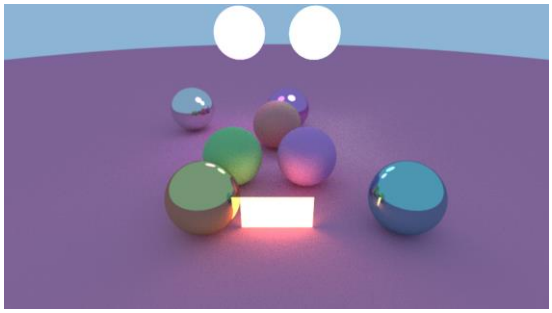
I rendered 2 dielectric spheres, which have different index of refractions. The one on the left has the index of refraction of 1.33 (water) and the one on the right has the index of refraction value of 2.417 (diamond). As the index of refraction increases, light refracts in a wider angle.



Task 6 [Required]: Lights (15Pt)

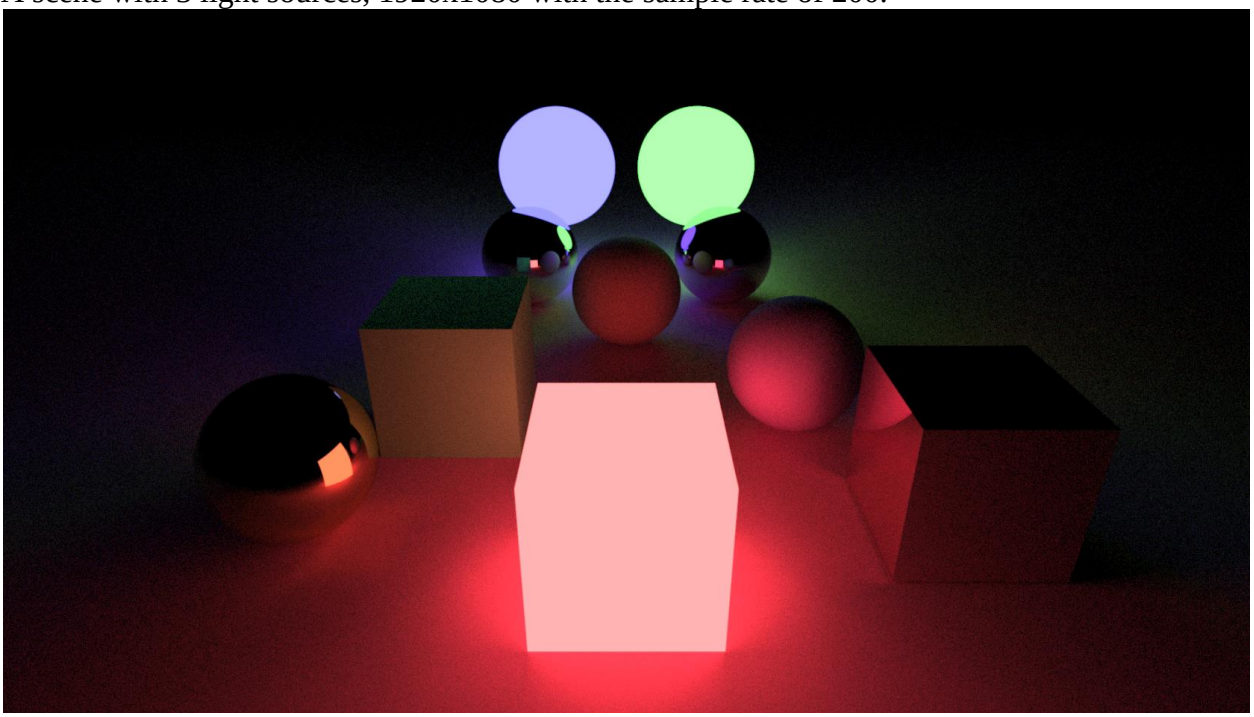
I used 2 renders to show the emissive materials. First one has no ambient light, but 3 light sources. Second one has a background so the other objects show their real color very clearly. The light sources are colored, also some objects are metal while others are Lambertian.





Task 7 [Optional]: Let's get creative! (10Pt)

A scene with 3 light sources, 1920x1080 with the sample rate of 200.



Task 8 [Required]: Questions (10Pt)

Answer the following questions:

- **The approximate time complexity of ray tracing on models with triangles.**

  O(px . log(n)), px being the number of pixels, and n being the number of triangles.

- **What is the difference between preprocessing and computing the image? Why?**

Preprocessing converts the data to an input which is easily readable by another program (most commonly a compiler) however it doesn't make sense to us humans. Computing the image (rendering) converts the data provided by the preprocessing operation and calculates the image so that we can read.

- **What are the critical parameters for the ray tracer algorithm's performance?**

Depth is the one of the most important aspects of ray tracing. It is the maximum number of times a ray can be reflected. As it increases, ray tracing would cost more time and space since recursive algorithm would be called and it would take much more time to compute.

The pixel count is also very important. As the number of pixels increases, the more computing has to be done. Keeping it low means less computing, faster render time.

The number of objects in the scene is another important aspect of ray tracing. Whenever a ray hits an object, a new calculation has to be made.

- **Imagine you are a systems engineer at Pixar. There is a new super-resolution in 6000×6000 pixels and you have to estimate the maximum rendering time per frame. Assume that the scenes are static, so you are not going to spend any CPU on animation, scene hierarchy, etc. The average scene has 500 objects with a total of 5.000.000 triangles to check intersection with.**

We have a total of 36.000.000 pixels to render per frame. Assuming we use anti-aliasing, we multiply our pixel count with sample rate. We need to send that many rays in order to render our frame. If we have 500 objects with a total of 5.000.000 triangles, that means we need to make different calculations for the materials and shades of those objects too. We don't know anything about the computing power of our machine; but to reach the maximum rendering time per frame, we need to put all of our objects in the field of view of the camera. Reflecting objects and high resolution textures would increase the maximum rendering time per frame even higher.