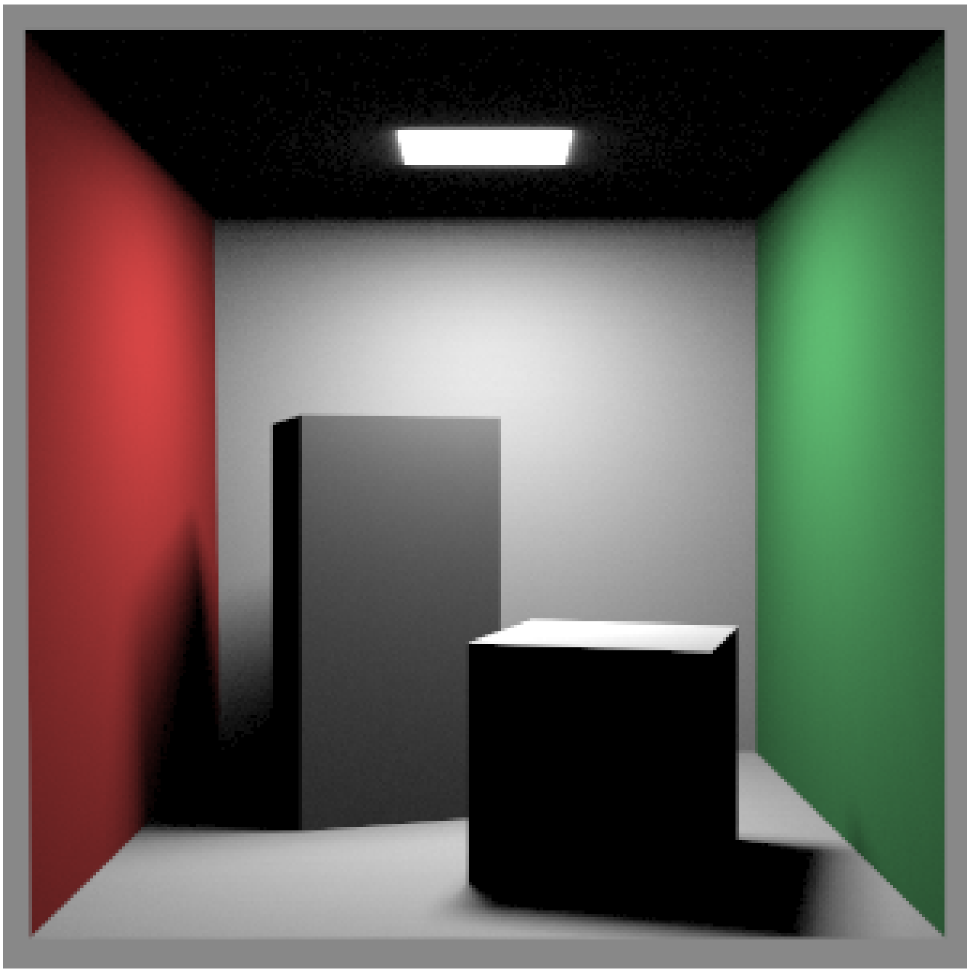
Report: Computer Graphics **Xucheng Chen**
COMS 4160 xc2360
Prof. Zheng Date: 02/23/2017

Rendered images

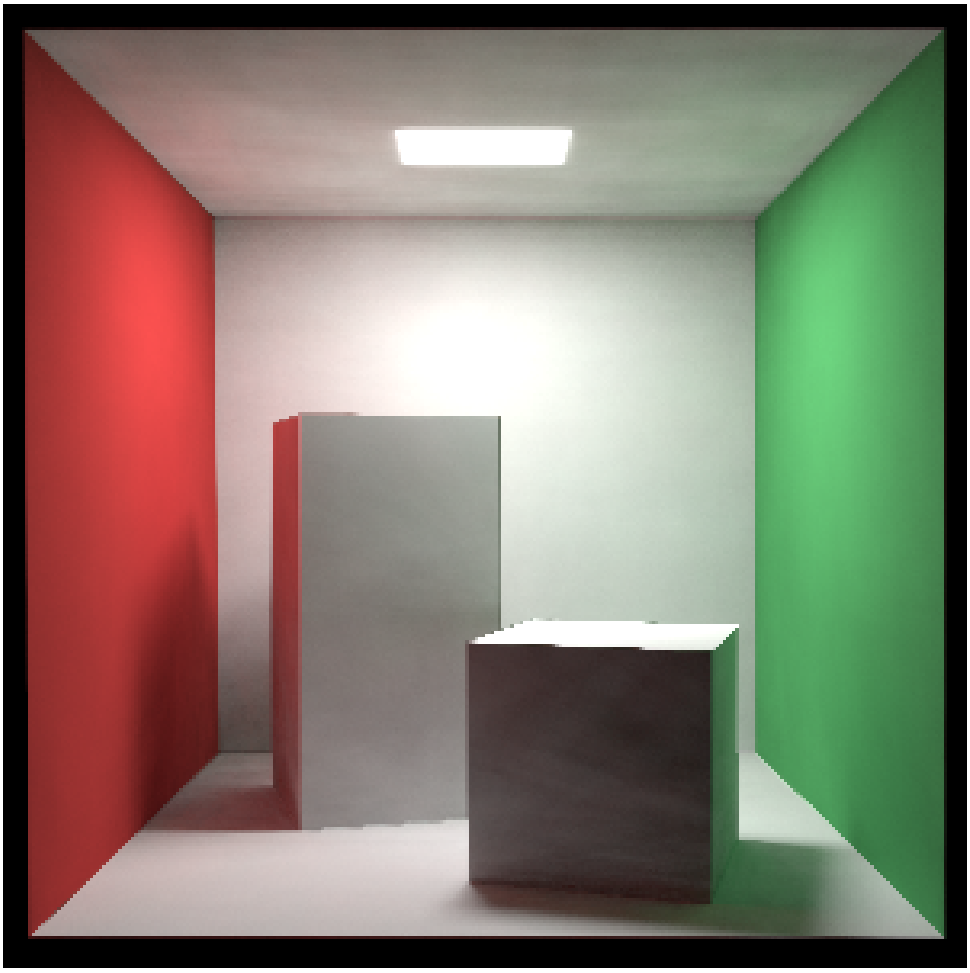
0.1 Direct Illumination

The direct illumination works fine for me and the resulting direct rendered image is direct-result.png and the original .xml file for it is cbox-direct.xml with sampler number of 100.



0.2 Global Illumination

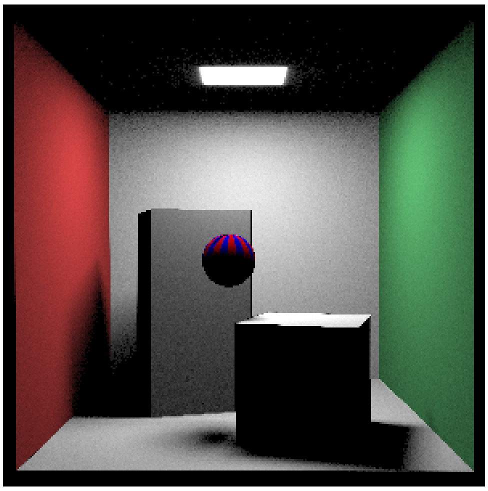
The global illumination takes me a lot of time in debugging. Although the overall algorithm is simple and concise, the implementation involves a lot of details. The rendered image is direct-result.png and the original .xml file for it is cbox-direct.xml with sampler number of 100.



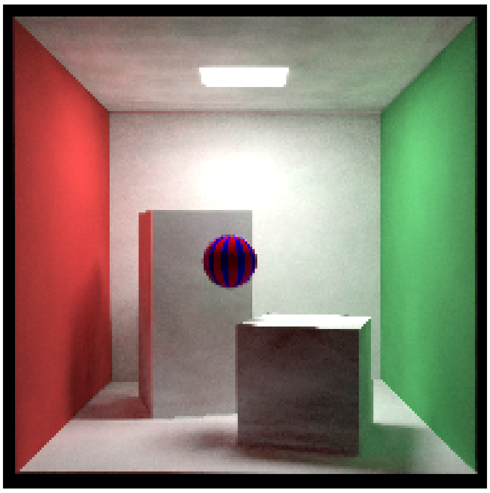
0.3 Creative Scenes

For the creative scene, I have set the image size to 200x200, the number of samples to 100 and the depth limit to 7 for the global illumination. If you want to change the texture type of objects, just simply change the variable texture in class Texture of package Material. The scene 1 use cbox-global-scene1.xml to render and the scene 2 use cbox-global-scene1.xml to render.

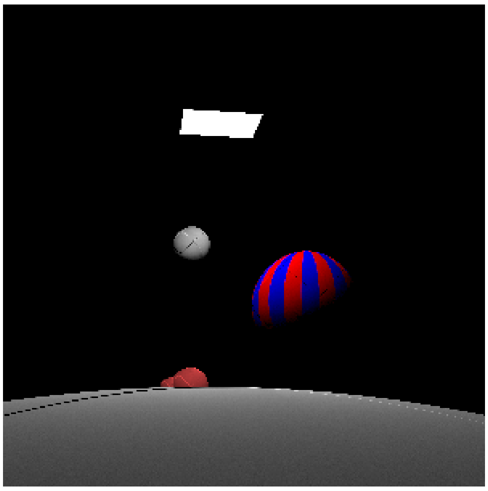
0.3.1 Scene1: direct illumination



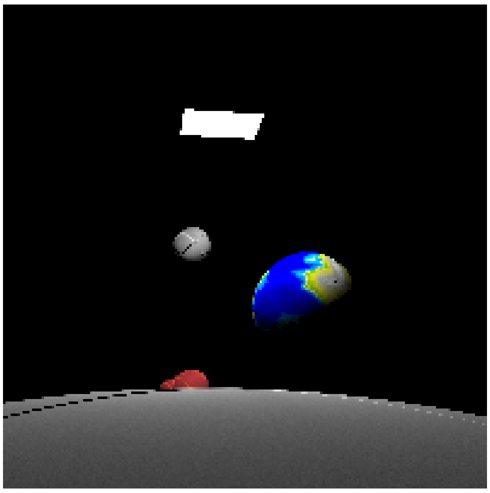
0.3.2 Scene1: global illumination



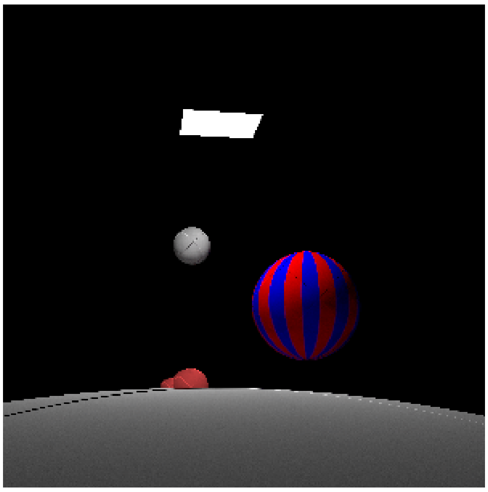
0.3.3 Scene2: direct illumination



Below is the direct illumination image of scene 2 using image texture, which takes very long time to render.



0.3.4 Scene2: global illumination



External resources

There are some packages called libnoiseforjava, which is the java wrapper of libnoise on github. I used these packages to generate the image texture picture, which is called noise.png.

Bonus extension

As I mentioned earlier, I have successfully map the texture on objects by creating a Material class named Texture.java, but it only works for spheres. The result is shown in "cbox-direct-image-texture.png".

Reminder

If you try to do image-texture rendering, you should wait about half an hour, while the others are quite faster. I strongly recommend using procedural texture instead.