Programming Assignment 4

COMPUTER GRAPHICS

Submission

Deadline: 23:59:59, Sunday, June 19th , 2020 (KST, +0900)

- Github server clock

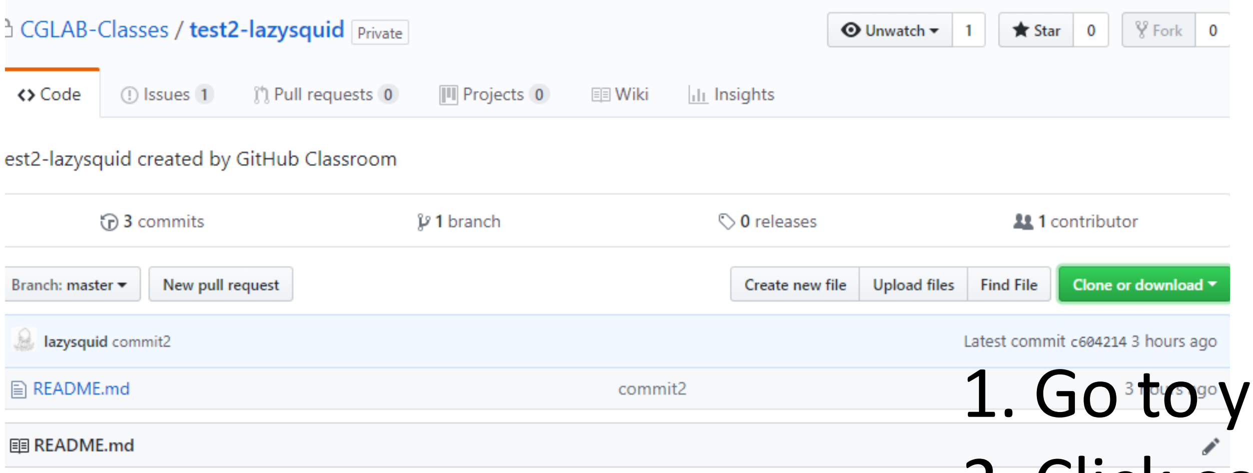
To submit your assignment, you **must** do two things. **Both of them must be done BEFORE deadline.**

1. You should push your commit to your assignment repo before deadline.
   -Obviously, e- mail submission is not accepted
2. You should comment the last commit (before deadline) id (SHA-1 hash) in github issue board. (See next slide)

The last commit BEFORE dead line will be considered as submitted assignment.

- Github server will track this for me.
- Timestamp in your commit (local time) will be igrnoed. (I will use github server timestamp instead)

Commenting Commit ID 1/2

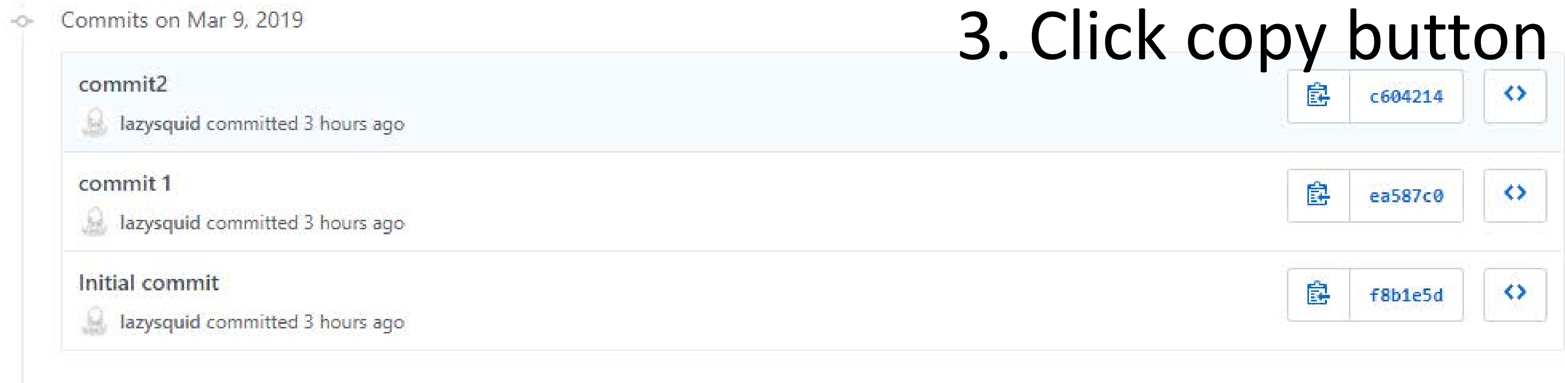


est2-lazysquid created by GitHub Classroom

lazysquid commit2 — Latest commit c604214 3 hours ago

README.md — commit2 — 3 hours ago

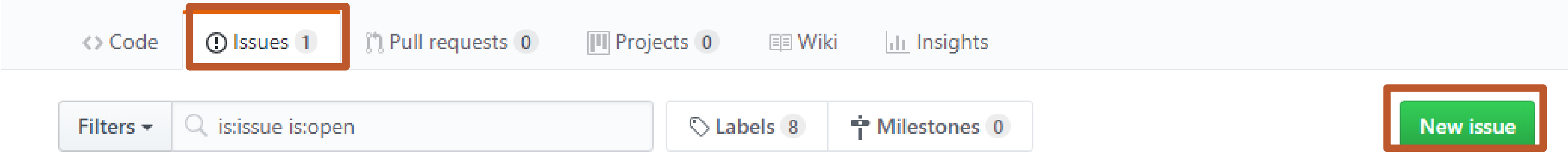
1. Go to your assignment repository
2. Click commits
3. Click copy button of your last commit



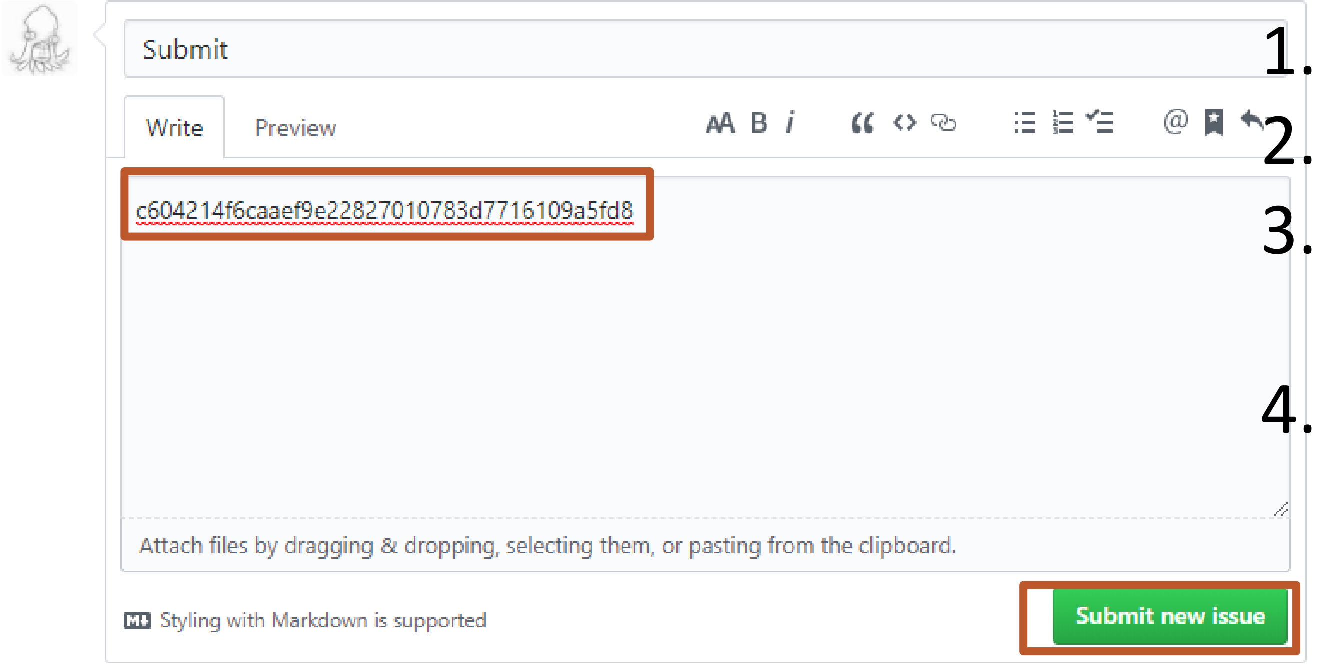
Commits on Mar 9, 2019

commit2 lazysquid committed 3 hours ago	c604214
commit 1 lazysquid committed 3 hours ago	ea587c0
Initial commit lazysquid committed 3 hours ago	f8b1e5d

Commenting Commit ID 2/2



is:issue is:open


Write
Preview
Attach files by dragging & dropping, selecting them, or pasting from the clipboard.
Styling with Markdown is supported
Submit new issue

Submit

c604214f6caaef9e22827010783d7716109a5fd8

1. Go to issue tab
2. Click “new issue”
3. Paste your lastest commit id (Ctrl-v)
4. Click “submit new isse”

Policy

In the following cases, your grade for this PA will be 0

- Late submission (Late push before deadline or Late last commit id comment on issue board)
- Build/execution failure
- Making public of your assignment repository
- If you tried to push your commit with force option(Tried to change history of remote server)

Your final grade will be "F"

- Copy

Neon

- A minimal ray tracer written in C++

- Base code for assignment 4

❖ Ray tracer

- Physically based rendering (pbrt)

  (https://pbrt.org/)

- Mitsuba renderer

  (https://www.mitsuba-renderer.org/download.html)

Structure

2 project

- **neon**

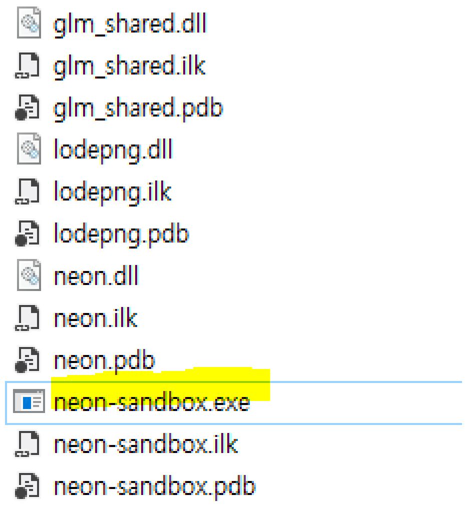
  - image.hpp; read and write images and so on
  - image.cpp
  - integrator.cpp; return light contribution of a path
  - integrator.hpp
  - scene.hpp ; sample direct light
  - scene.cpp

- **neon-sandbox**
  - test.hpp ; scene definition (objects, materials)
  - test.cpp
  - main.cpp ; rendering loop

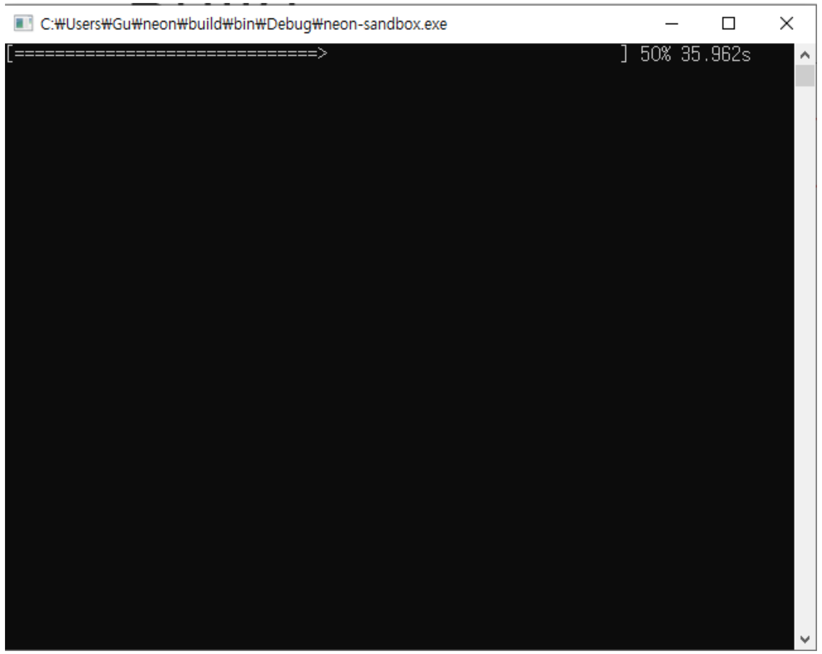
- sphere.hpp ; ray intersection test for sphere obj
- sphere.cpp
- intersection.hpp; record information of the hit point
- rendable.hpp
- ray.hpp ;
- material.hpp ;material properties such as scattering
- material.cpp
- utils.hpp
- utils.cpp
- camera.hpp; camera properties such as lens radius, fov
- blueprint.hpp

Build

- You can build Neon as same with glsekeleton (cmake)
- Run Neon, then a output will be a image (*.png)

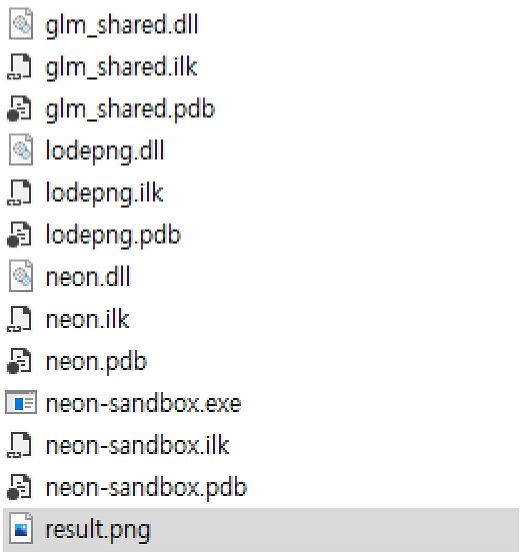


glm_shared.dll
glm_shared.ilk
glm_shared.pdb
lodepng.dll
lodepng.ilk
lodepng.pdb
neon.dll
neon.ilk
neon.pdb
neon-sandbox.exe
neon-sandbox.ilk
neon-sandbox.pdb



C:₩Users₩Gu₩neon₩build₩bin₩Debug₩neon-sandbox.exe

```
[============================>                              ] 50% 35.962s
```



glm_shared.dll
glm_shared.ilk
glm_shared.pdb
lodepng.dll
lodepng.ilk
lodepng.pdb
neon.dll
neon.ilk
neon.pdb
neon-sandbox.exe
neon-sandbox.ilk
neon-sandbox.pdb
result.png

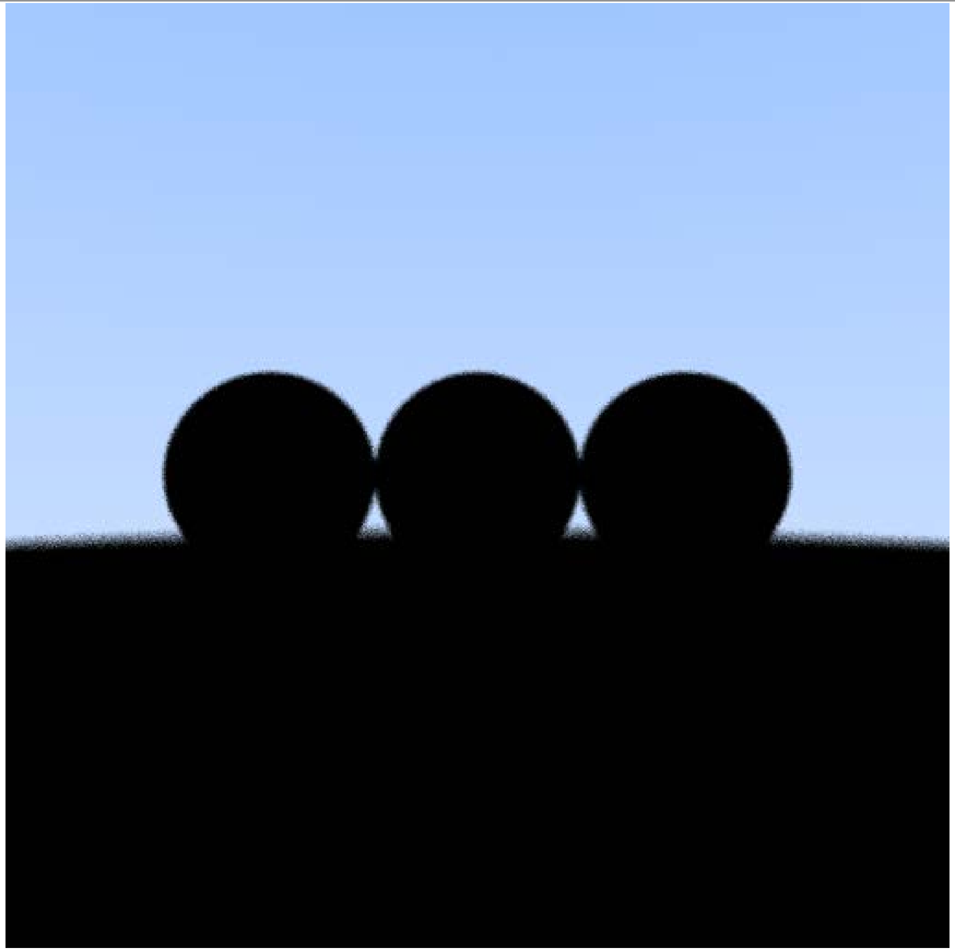
Task List

1. Materials (10 Points)
   - Lambertian, Metal, Dielectric, Area light(Emissive material)
   - Implement **scatter** function in each material class
2. Antialiasing (5 Points)
3. Indirect lighting (5 Points)
   - Multiple bounces, depth > 10
4. Direct light sampling (5 Points)
5. Defocus blur (5 Points)
6. Report (10 Points)
   - For this time, you need to write detailed report.
   - Add teaser image whenever you add new features(e.g. complete your task) and explain about it

Initial Appearance

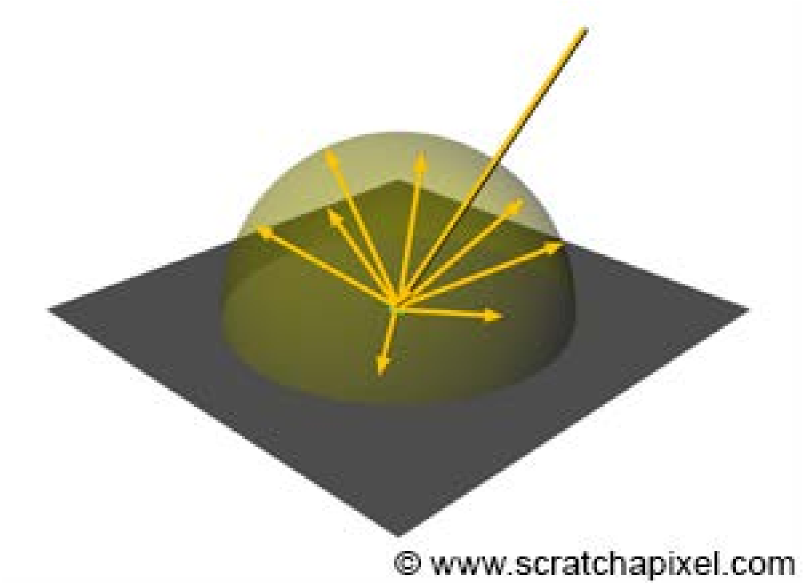
- Skeleton code: Neon renderer
- Unlike OpenGL project the result will be png file.

- output: *.png

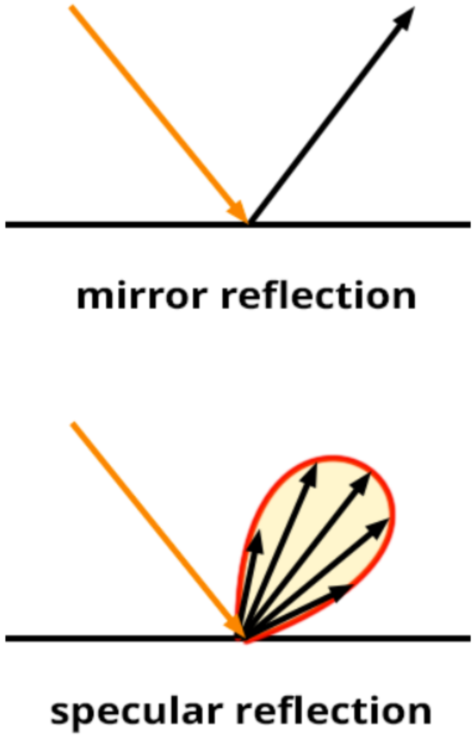


Materials

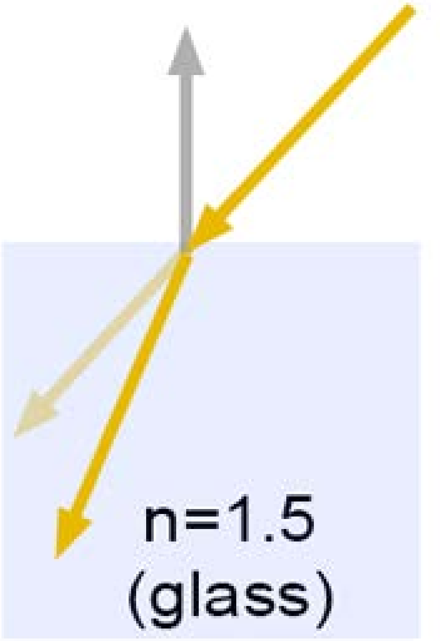
See **scatter** method in each material class



Lambertian (diffuse)



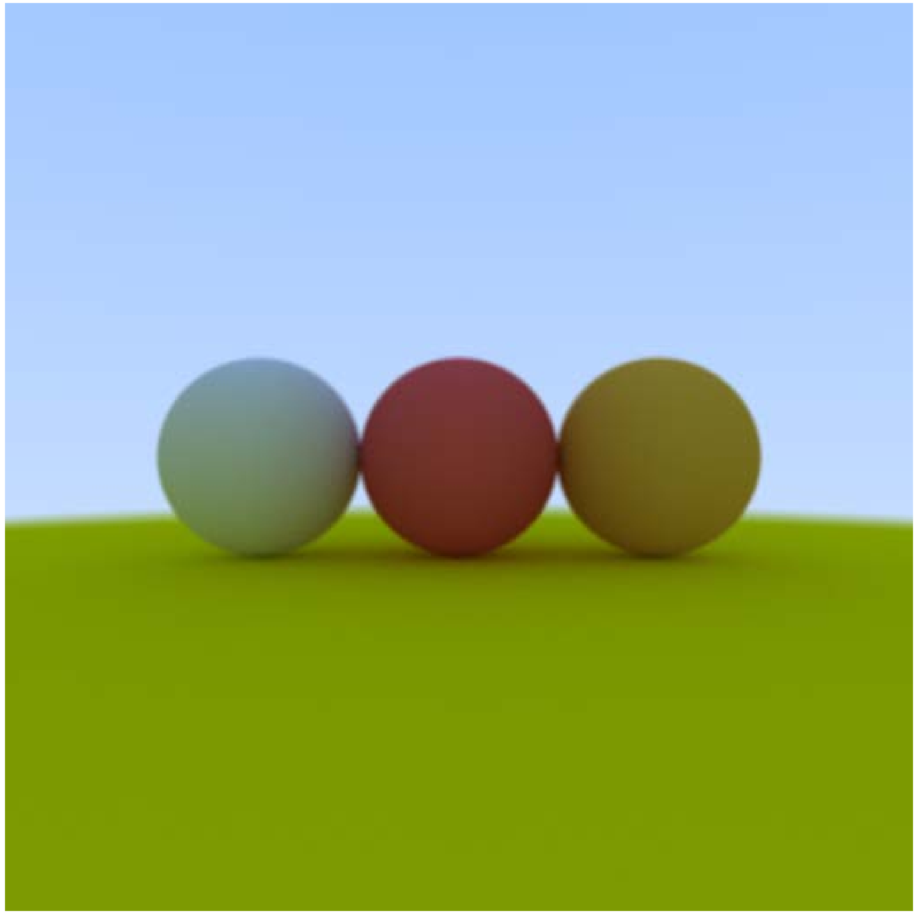
Metal(Mirror reflection with some randomness)



Dielectric

https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-shading

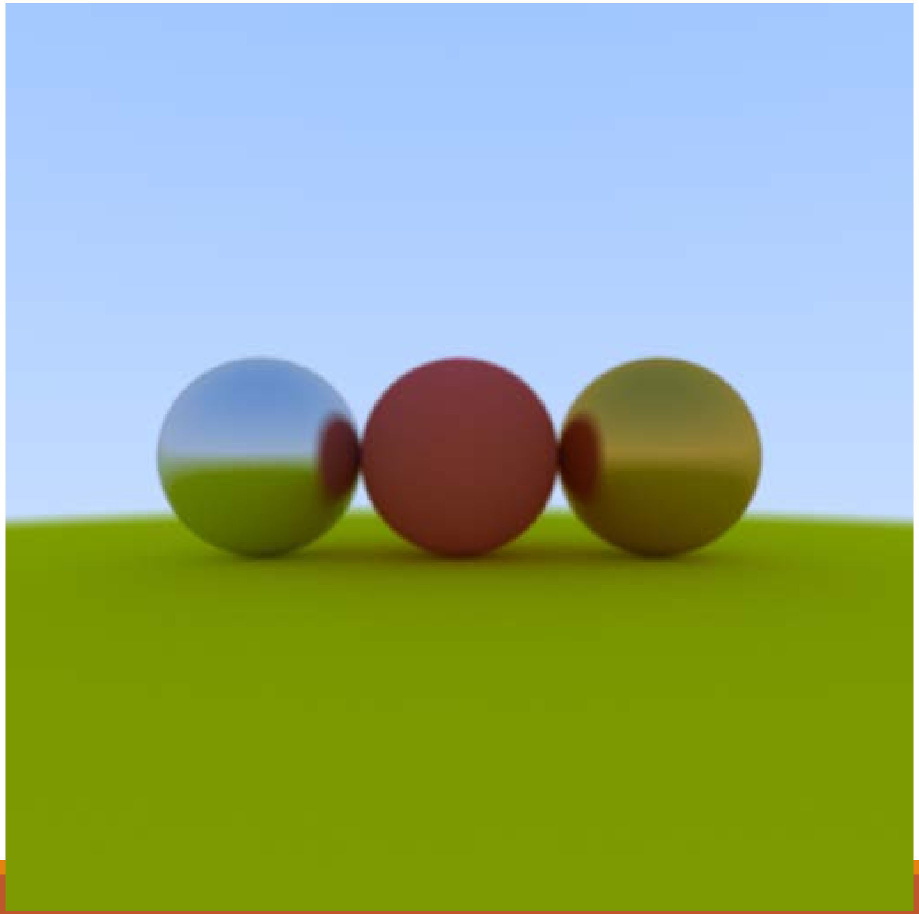
Materials

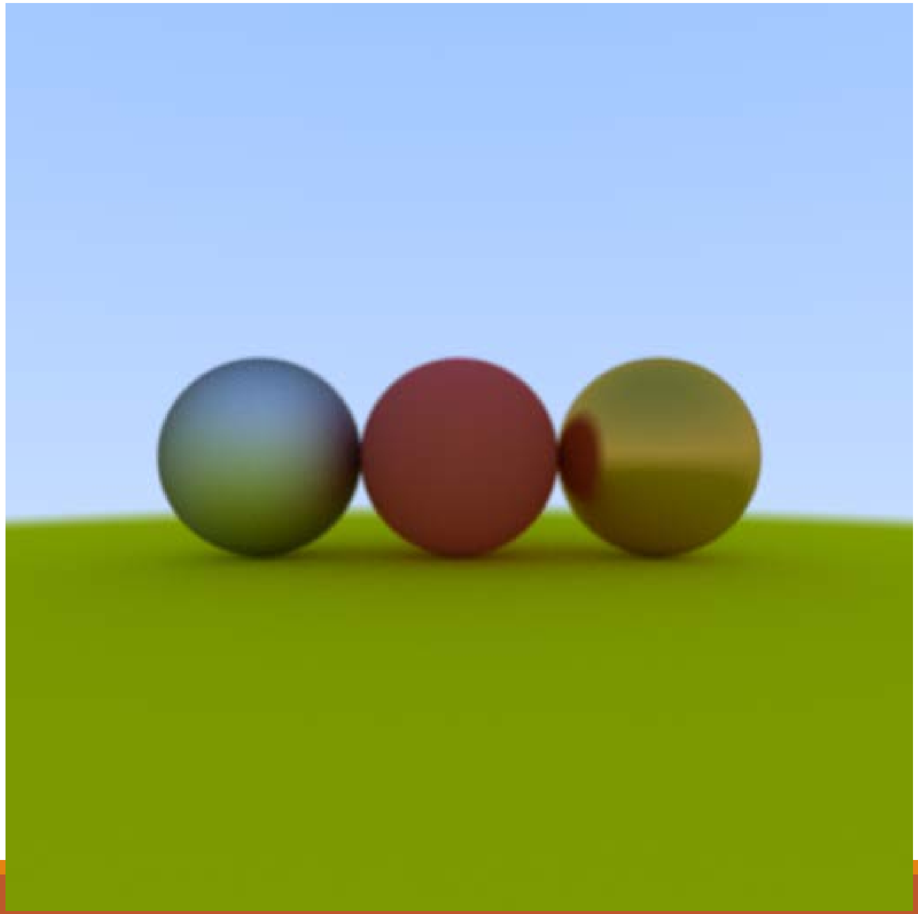


Lambertian

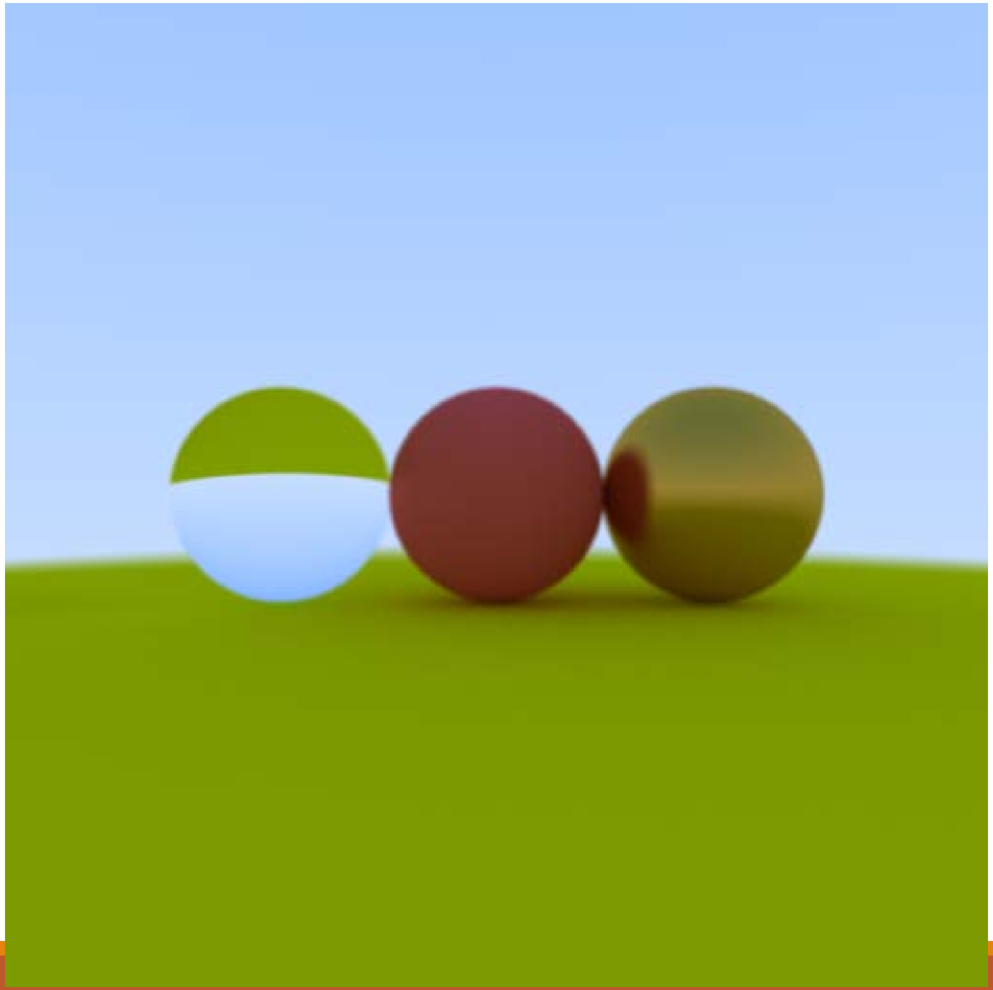
Materials

Perfect mirror vs metal (mirror with randomness)





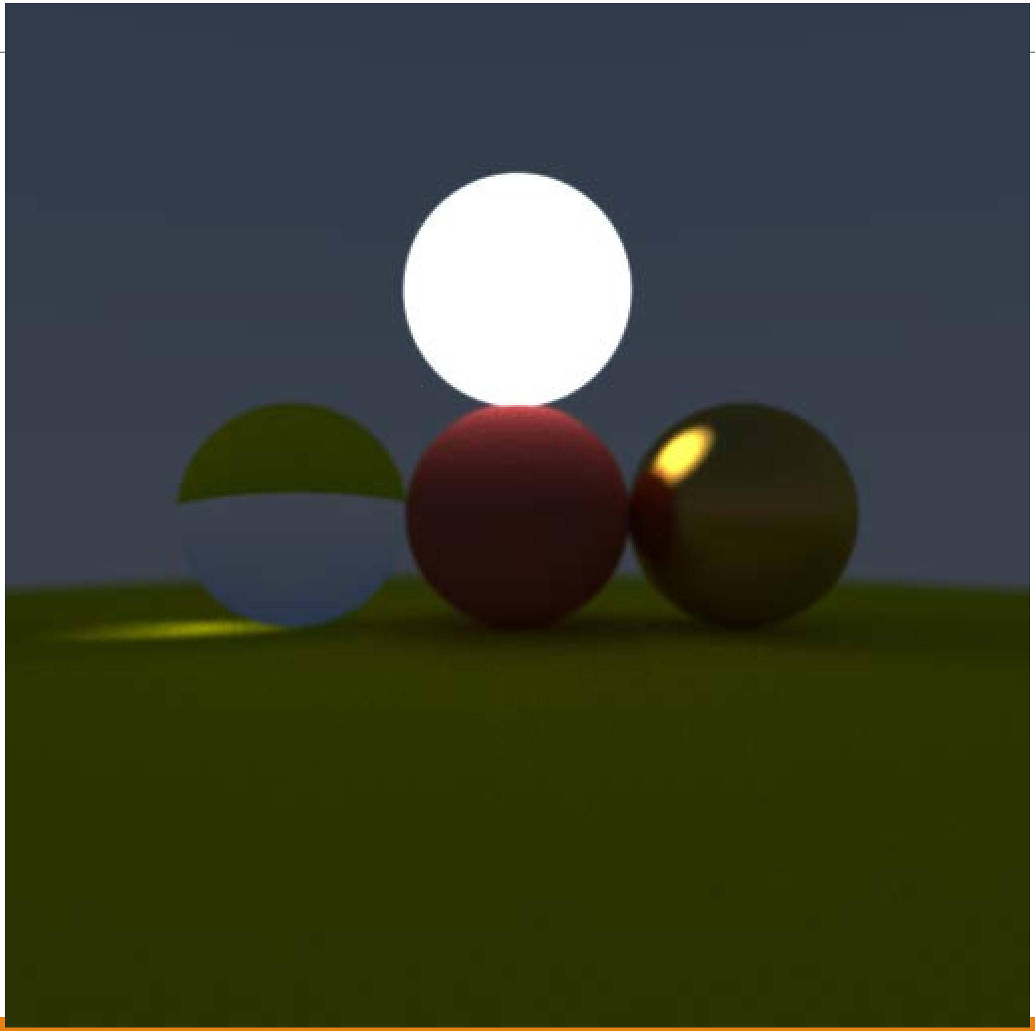
Materials



dielectric material

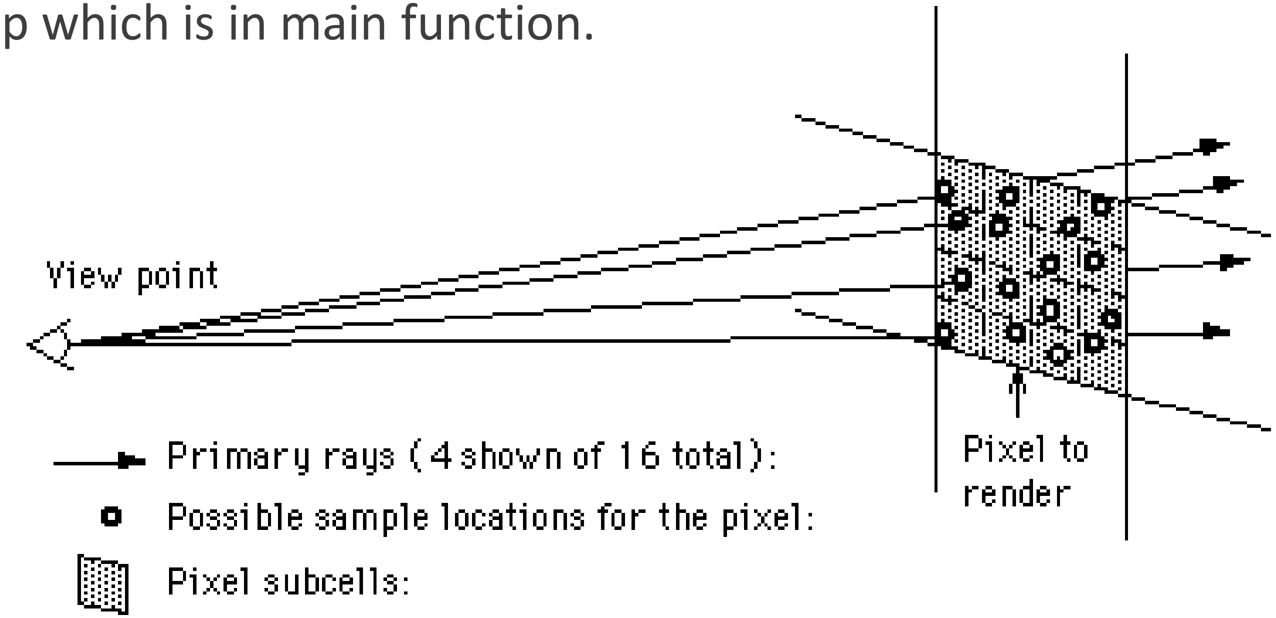
Materials

- Light ball
- Perfect glass ball
- Perfect diffuse ball
- Glossy metal



Antialiasing

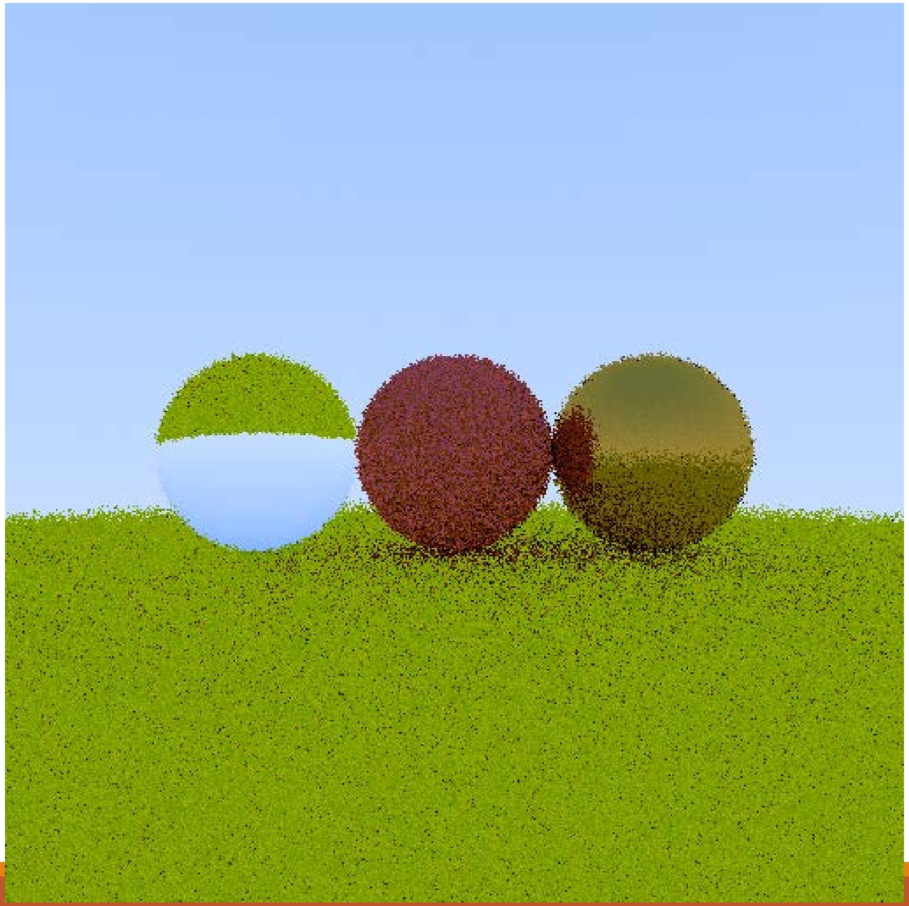
- Shoot multiple rays per pixel
- Final color will be average of those ray colors
- You can control this in rendering loop which is in main function.

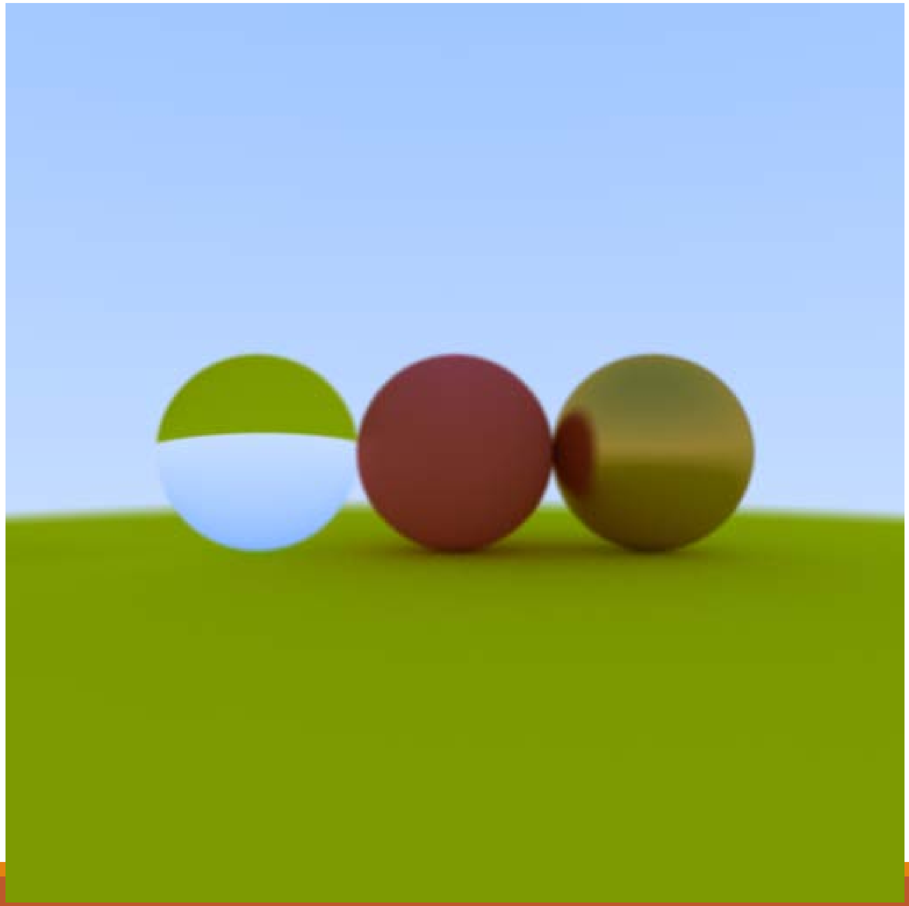

View point
Pixel to render
Primary rays (4 shown of 16 total):
Possible sample locations for the pixel:
Pixel subcells:

http://www.cs.montana.edu/~halla/cs525/intro.html

Antialiasing

1spp vs 1024 spp (samples per pixel)



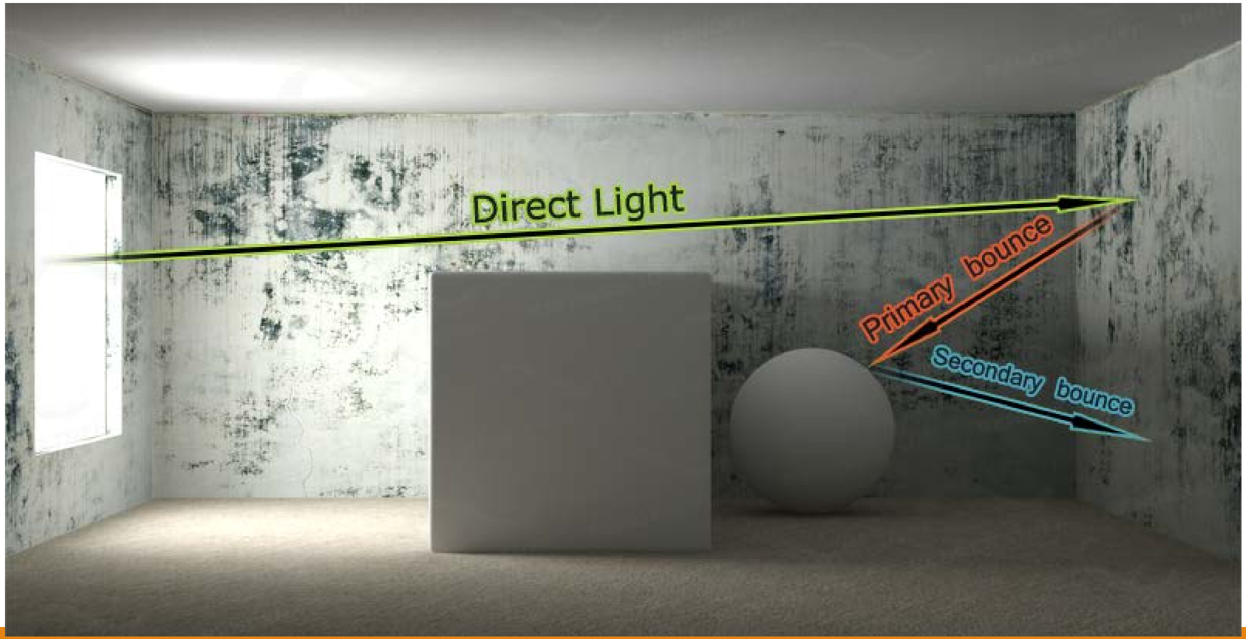


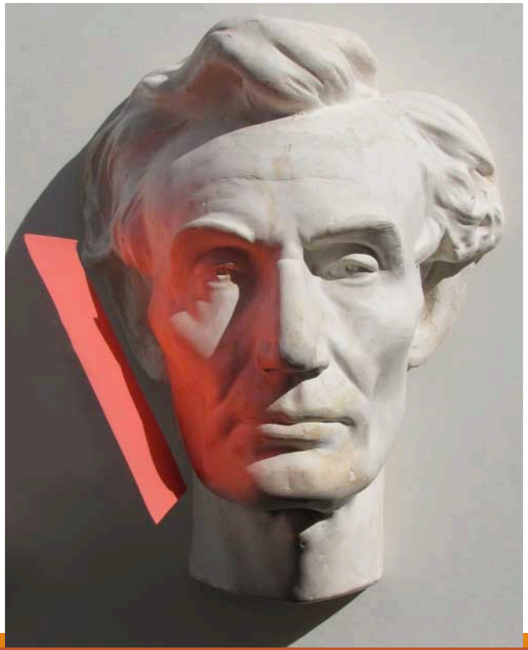
Indirect Lighting

Simulate multiple bounce of light.

You can see color bleeding(Diffusive interreflections) after this!

See **integrate** method in **integrator** class to control this behavior



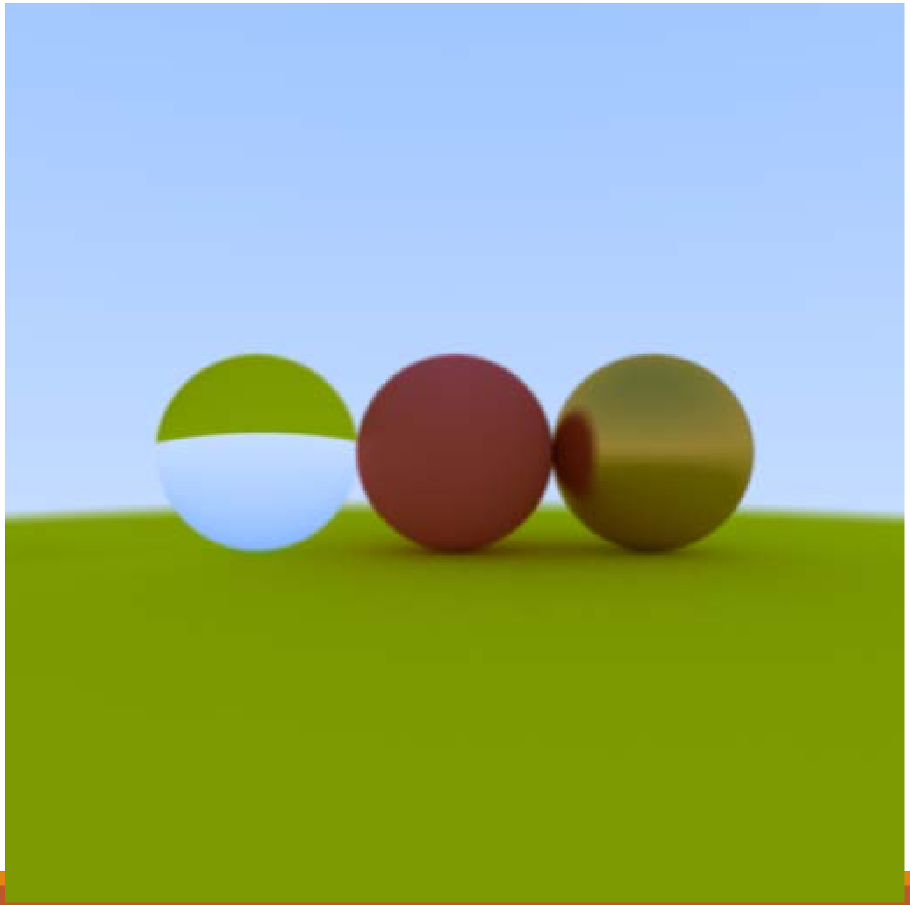


https://www.pinterest.co.kr/pin/362117626263103458/

http://gurneyjourney.blogspot.com/2010/05/color-bleeding.html

Indirect Lighting

See red color bleeding under the red sphere

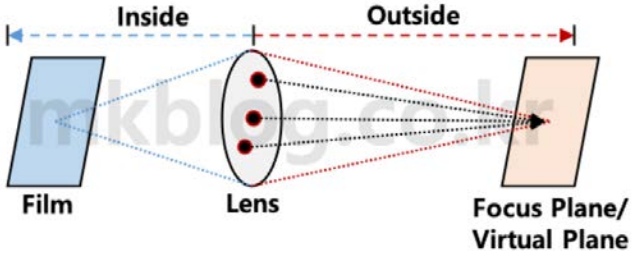


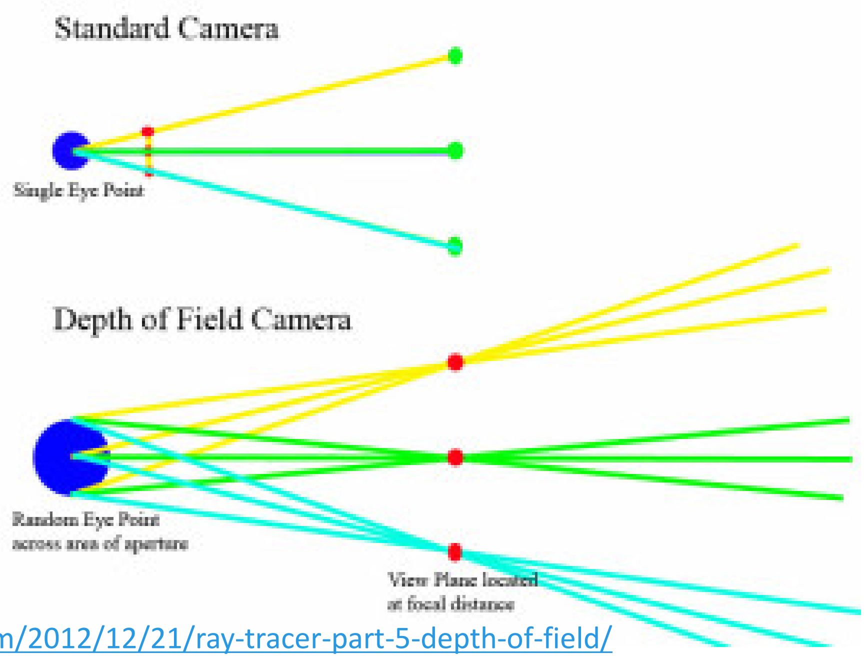
Defocus Blur

A.K.A Out focusing == Simulating lens effect

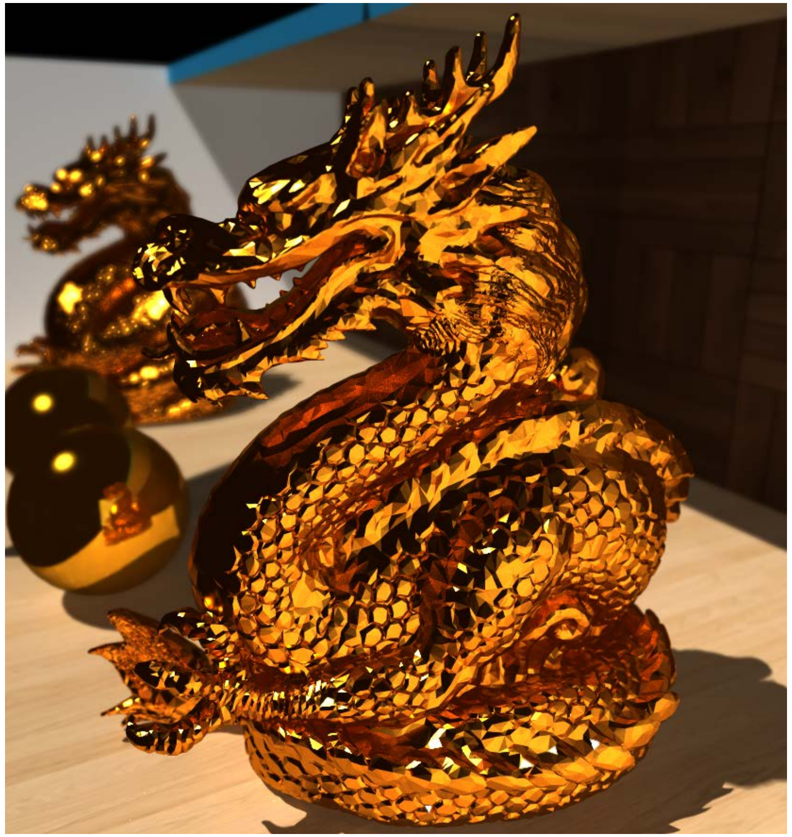
Generate random 2d point and add to ray origin.

See **camera** class to implement this



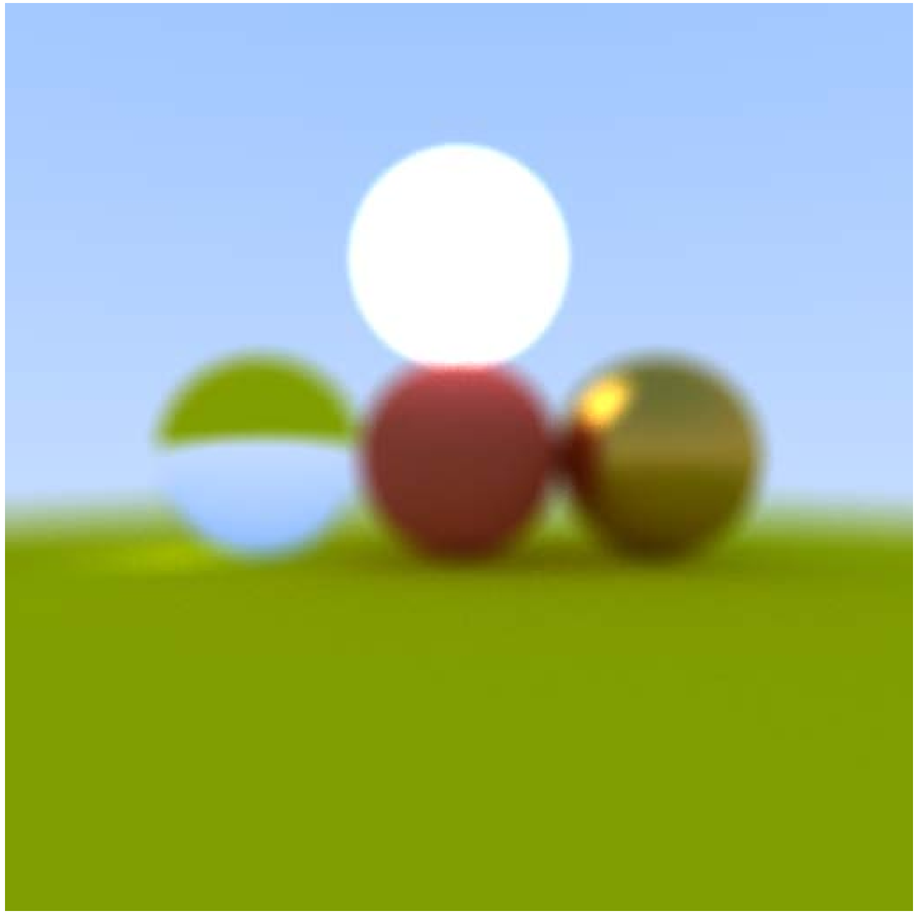


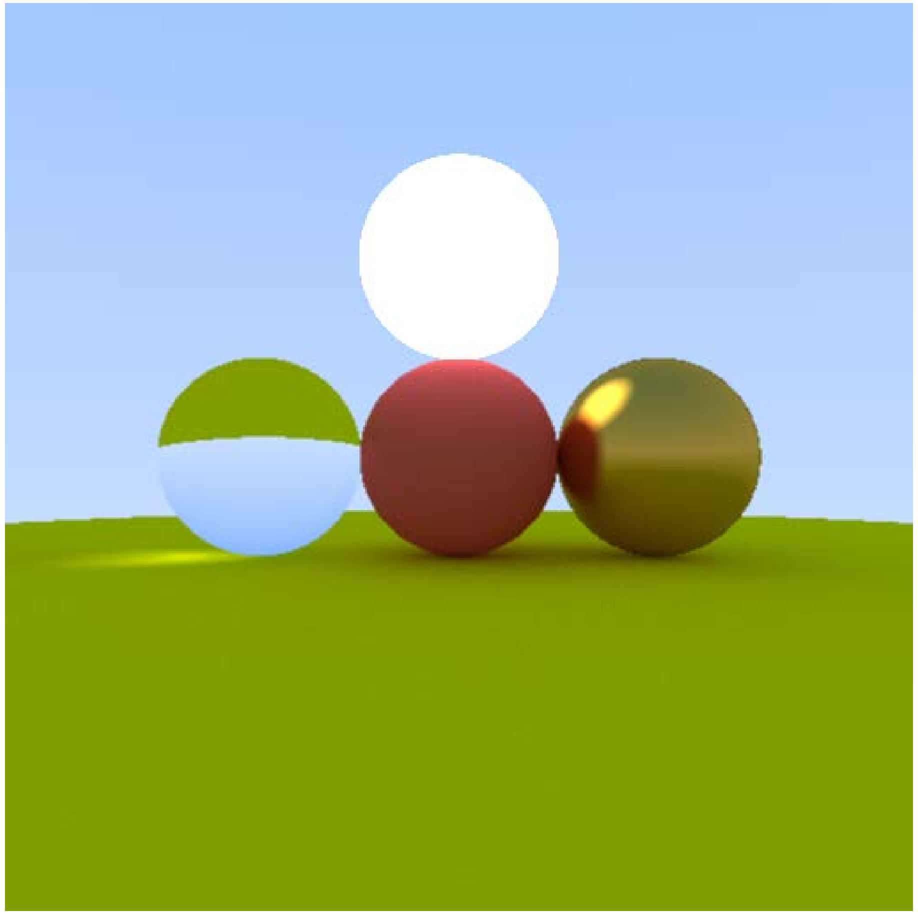
https://steveharveynz.wordpress.com/2012/12/21/ray-tracer-part-5-depth-of-field/



https://www.nebularender.com/gallery.html

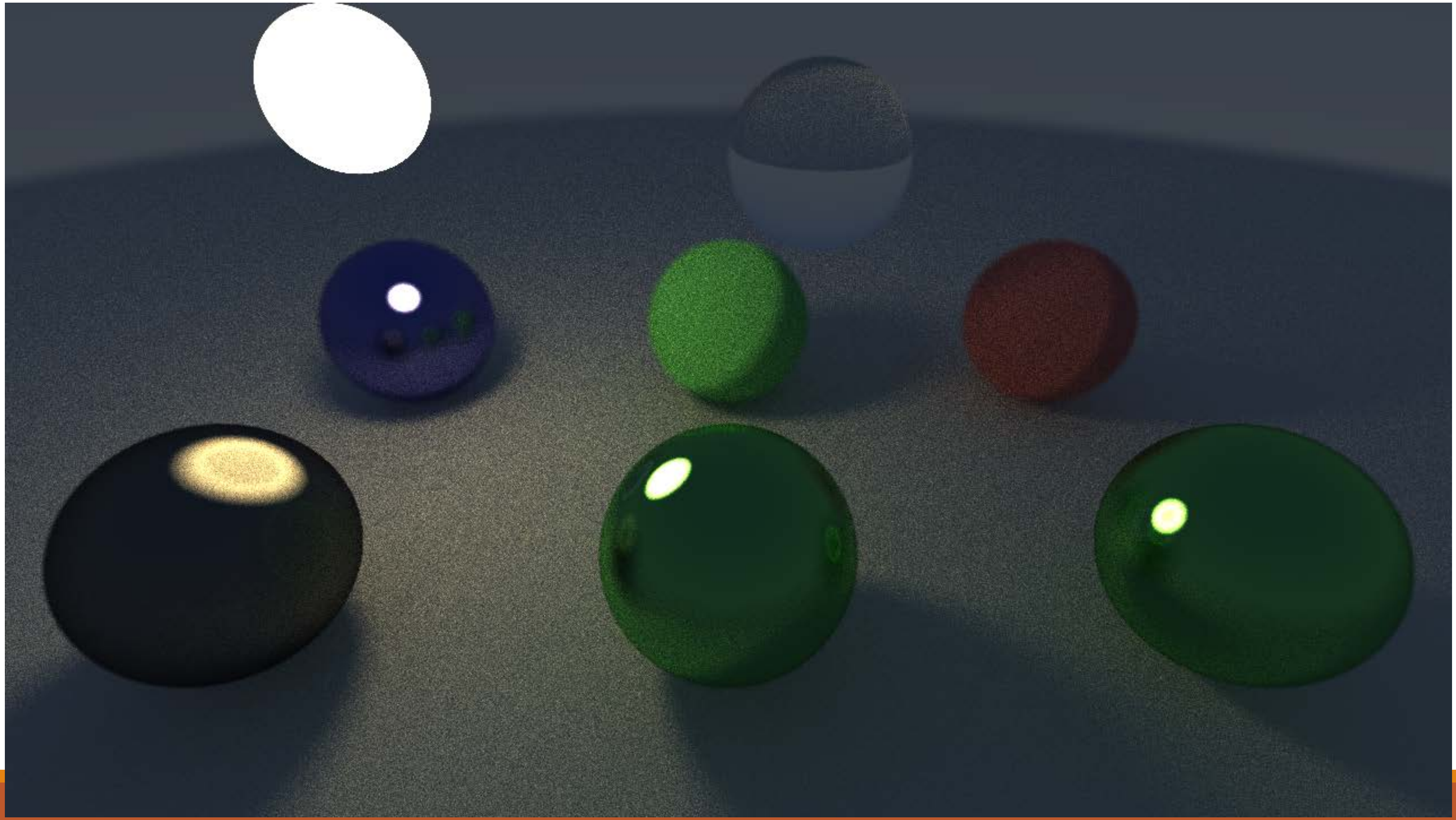
Defocus Blur





Direct Light Sampling

You can remove these noises if you are using direct light sampling



Scenes

To test the direct illumination, please use ‘testScene2’

```
// create scene
 std::shared_ptr<ne::Scene> scene = testScene2();
```

PA4 Link

1. Login to github
2. Go to following link – https://classroom.github.com/a/IpWatugl
3. Accept the assignment