README
[Assignment 1 - CSCI 2240]
Path Tracer

Description

This project contains code to perform Monte Carlo Path Tracing written in C++ and was built on top of the Stencil code provided to us.

Requirements

Qt (> 5.9.0)
Qt Creator (> 4.5.0)
OpenMP

Usage

Build :

- Using Qt Creator.
- Using CLI:

```
 cd Path_Tracer_2240
mkdir build
cd build
qmake -makefile ../path-stencil.pro
make -j4
```

Run :

- Using Qt Creator

  Set the following arguments in Qt Creator.

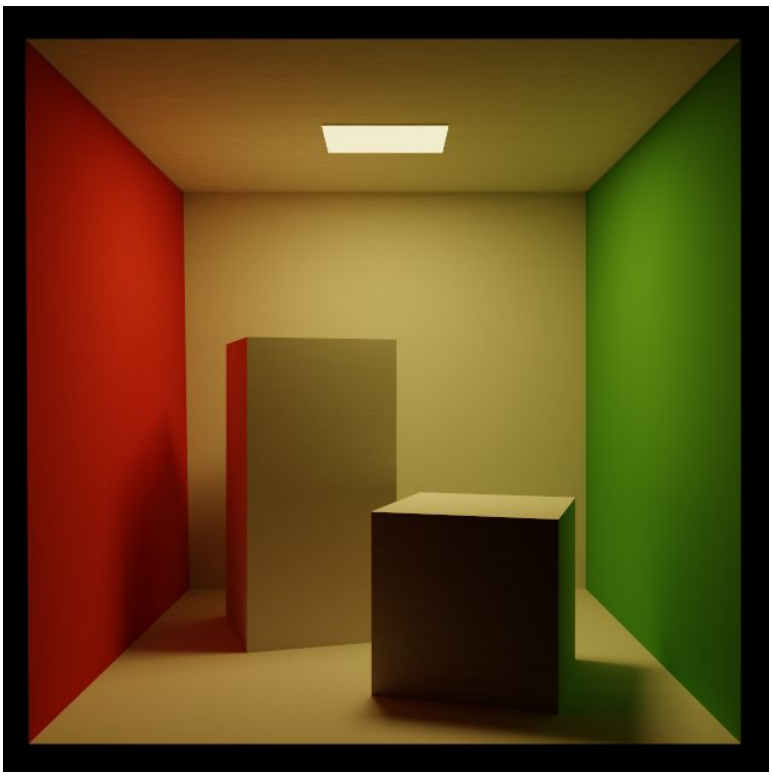
  # <path to xml file> <rendered image path> <number of samples> <image height> <image width>
  `../Path_Tracer_2240/example-scenes/CornellBox-Sphere.xml ./output.png 100 256 256`

- Using CLI :

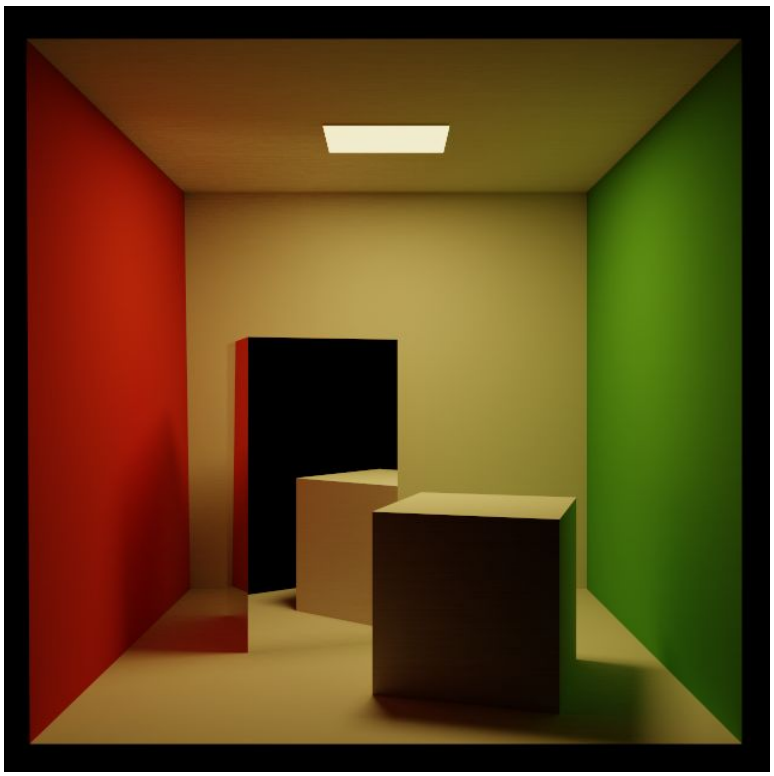
  ```
  ./path-stencil ../Path_Tracer_2240/example-scenes/CornellBox-Sphere.xml ./output.png 100 256 256
  ```

Implementation

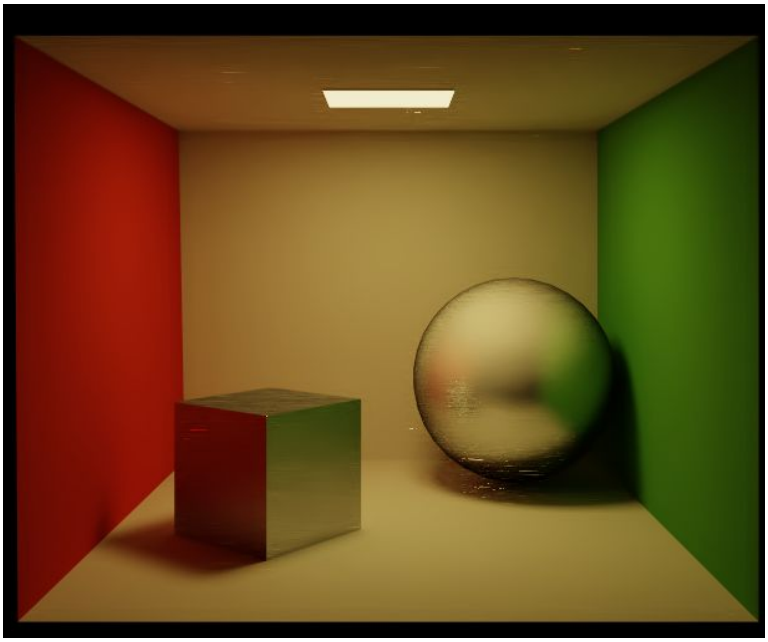
- Four basic types of BRDFs ✔



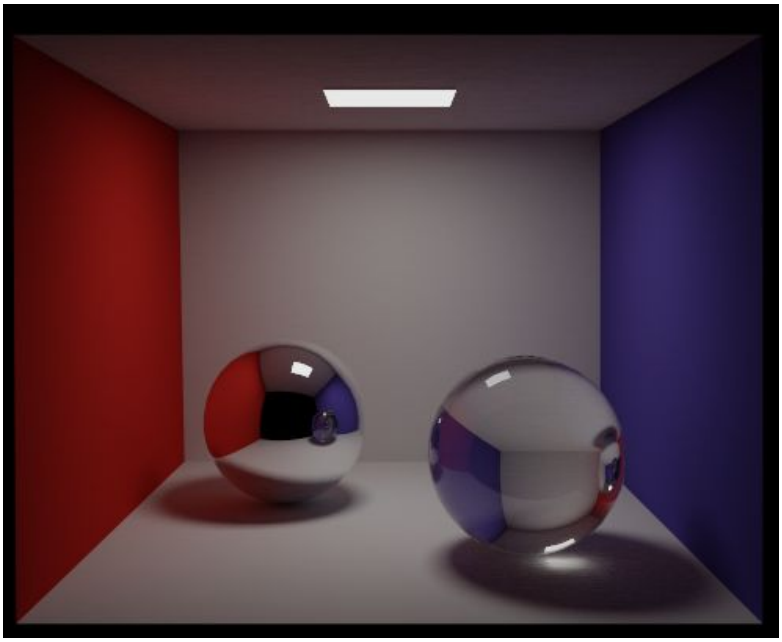
Diffuse [640 x 640] [2000 Samples]



Mirror [640 x 640] [2000 Samples]

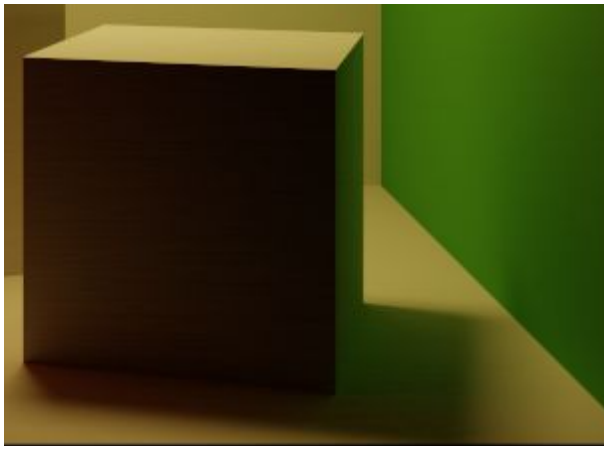


Glossy [640 x 640] [3000 Samples]

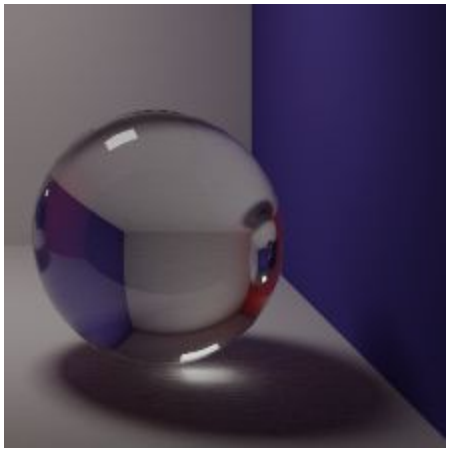


Refraction [640 x 640] [1500 Samples]

- Soft Shadows and Indirect Illumination ✔



Soft Shadows and Colour Bleeding

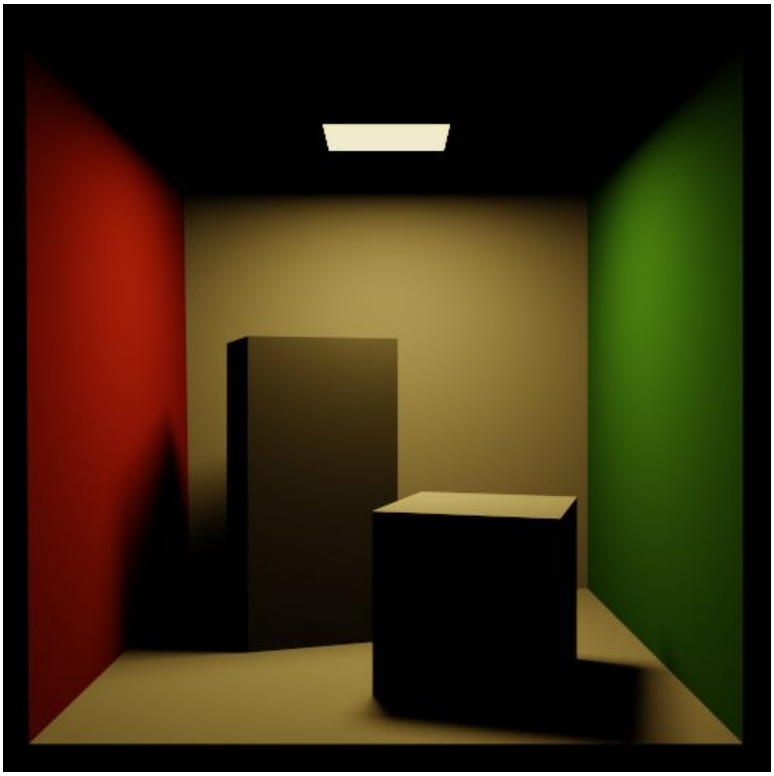


Caustics

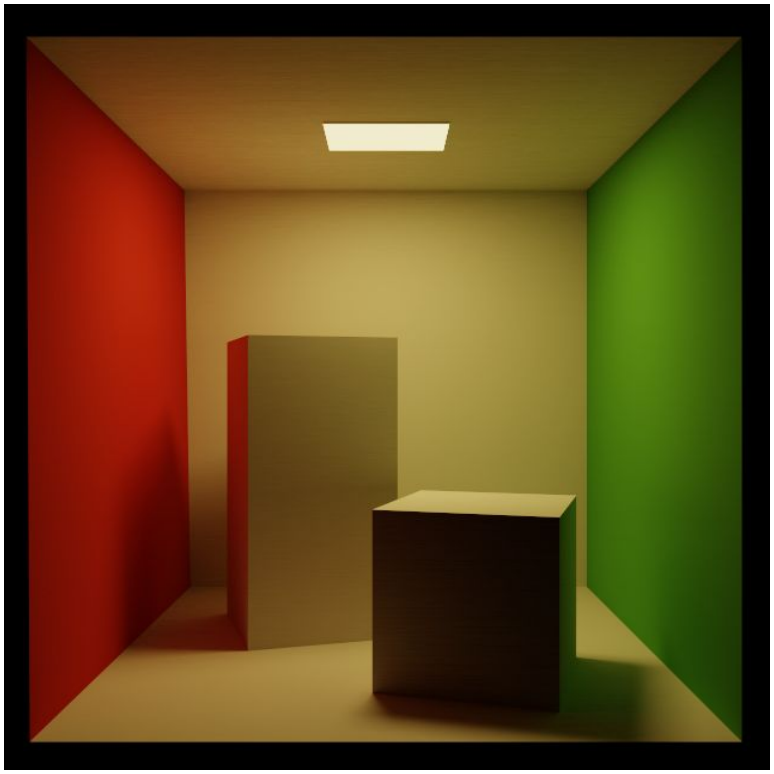
- Russian Roulette path termination ✔

- Tone Mapping ✔

- Event splitting ✔



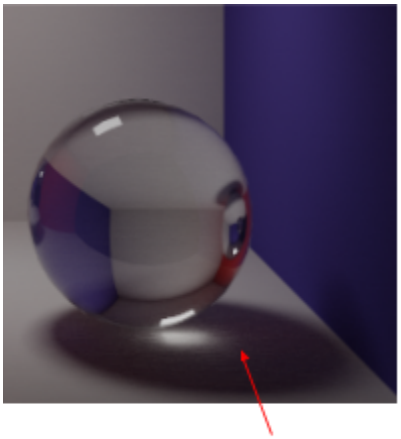
Only Direct Lighting [640 x 640] [400 Samples]



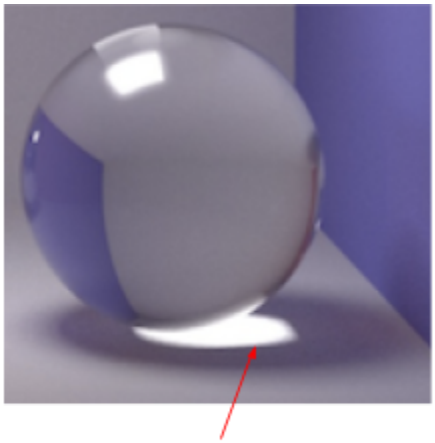
Full Global Illumination [640 x 640] [2000 Samples]

Extra Features

- Attenuate refracted paths ✔



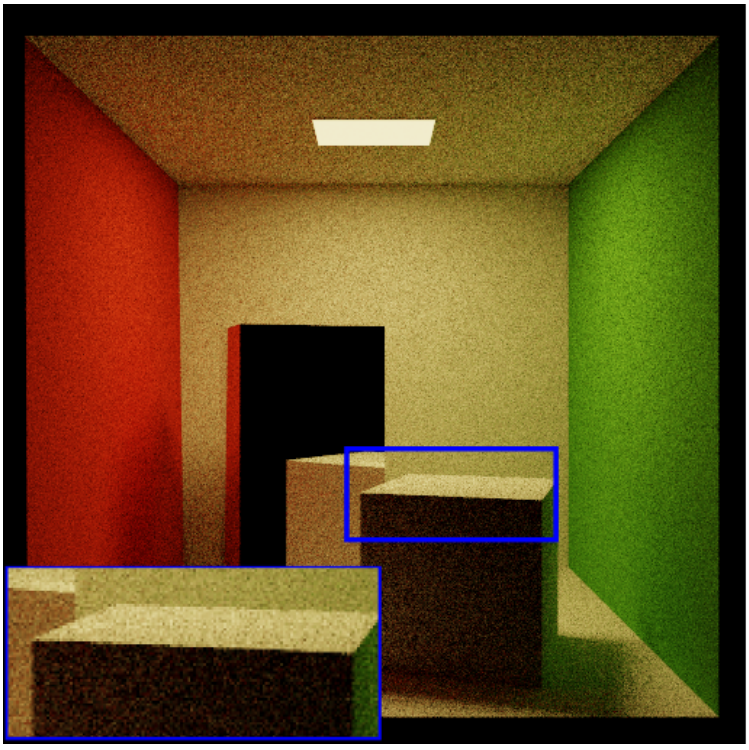
With Attenuation: Refracted rays lose their intensity. [2000 spp] [*This is my render*]



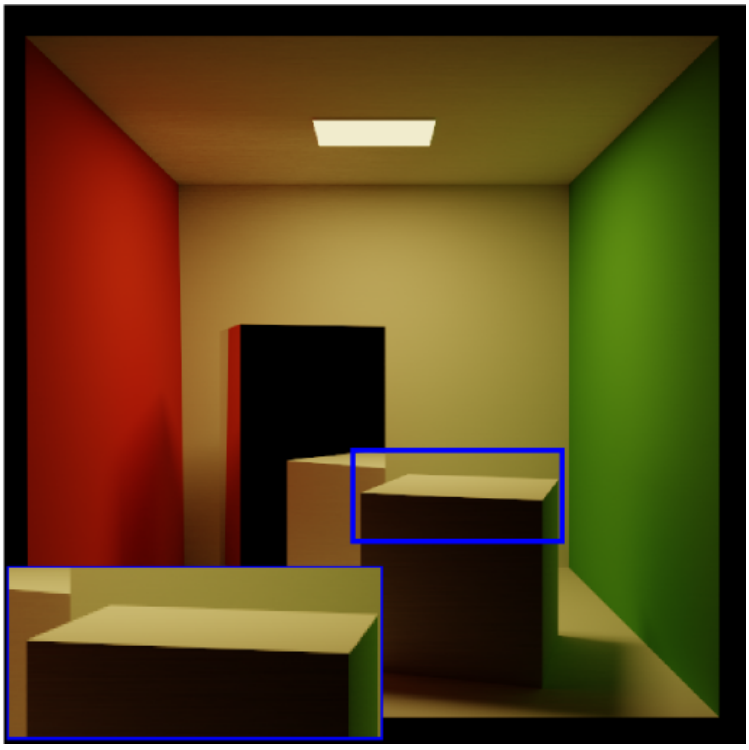
No Attenuation. Source: Henrik W Jensen [*This is NOT my render*]

Reference : https://www.scratchapixel.com/lessons/3d-basic-rendering/global-illumination-path-tracing

- Importance Sampling ✔



Uniform Sampling [640 x 640] [2000 Samples]

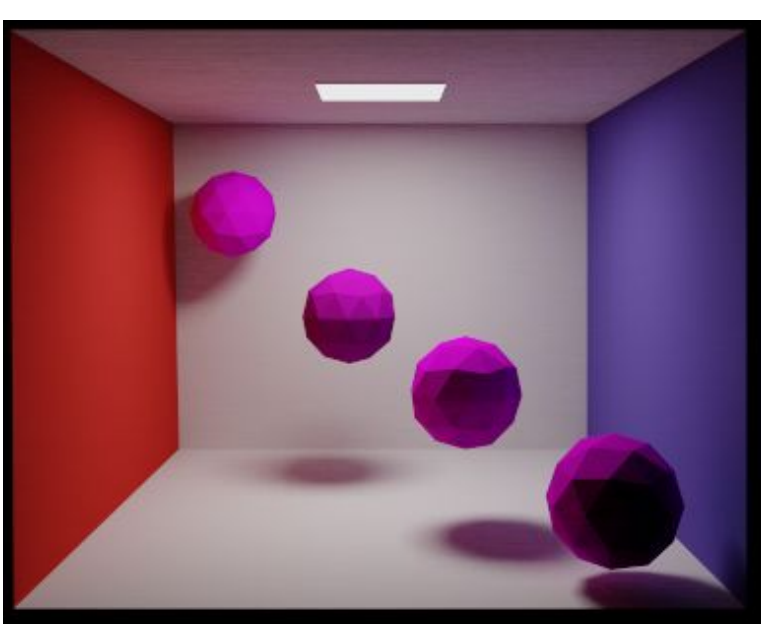


Importance Sampling [640 x 640] [2000 Samples]

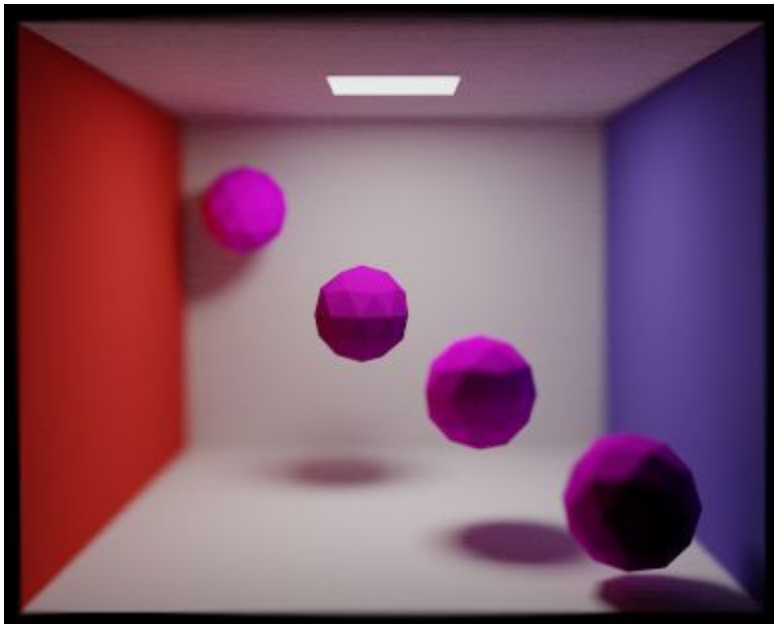
With importance sampling, the path tracer converges within 2000 samples per pixel, while the naive uniform sampling technique still produces noise for the same samples per pixel.

Reference: https://www.tobias-franke.eu/log/2014/03/30/notes_on_importance_sampling.html

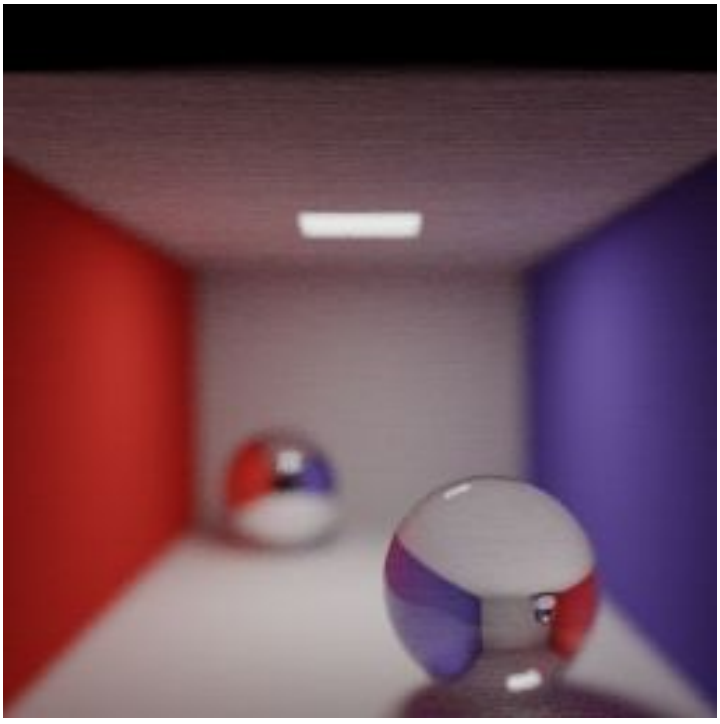
- Depth of field ✔

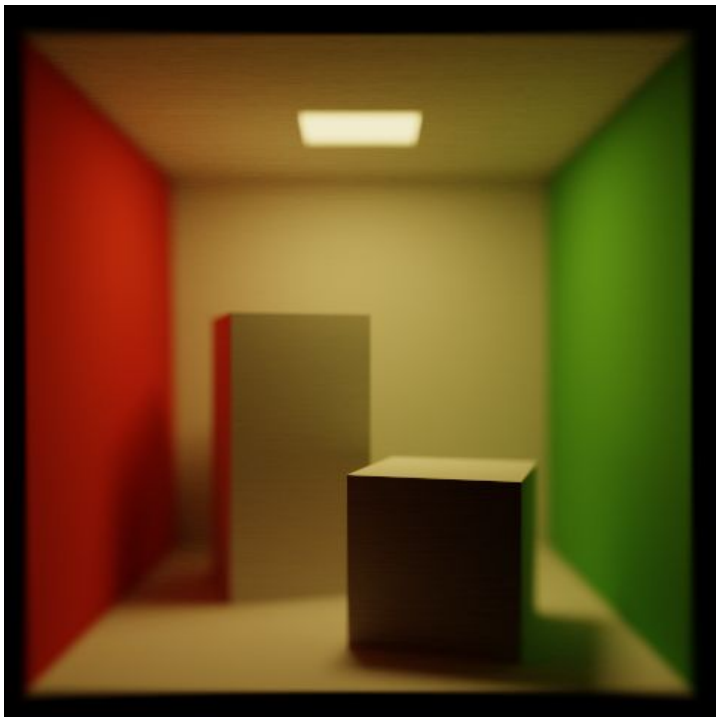


Normal



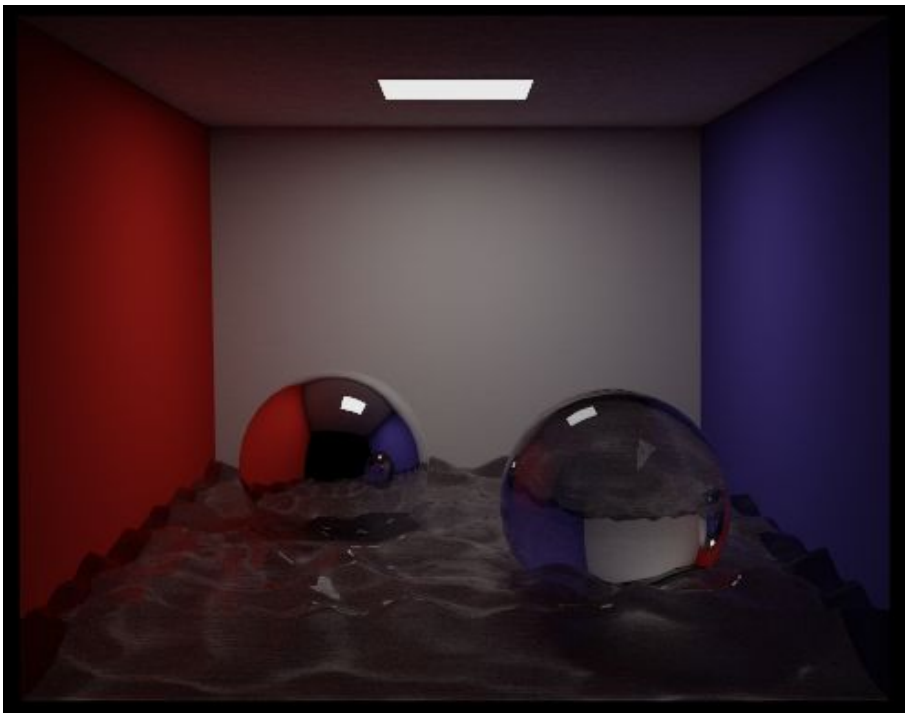
Depth of Field applied



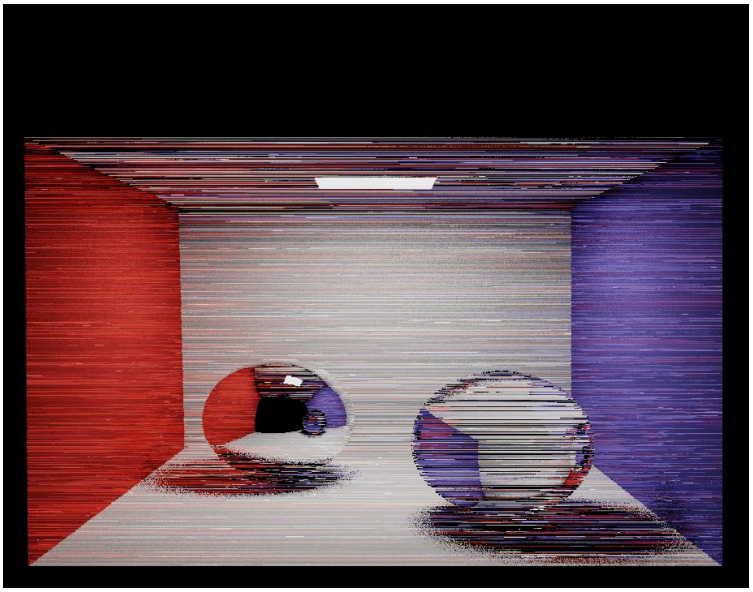


Implemented in function *depthOfField()* in pathtracer.cpp

Other Scenes



Bloopers



I have no idea what happened here.

Accidentally modified the diffuse material.