HW06 Readme – Ruoyu Fan

I have done:

- 2 Creating your own scenes to render (30 - 50 points)
- 3.4 Progressive rendering (80 points)
- 3.7 BVH integration and the surface area heuristic (50 points)
- 3.9 Depth of field (40 points)
- 3.11 Other interesting features: Interactive Rendering

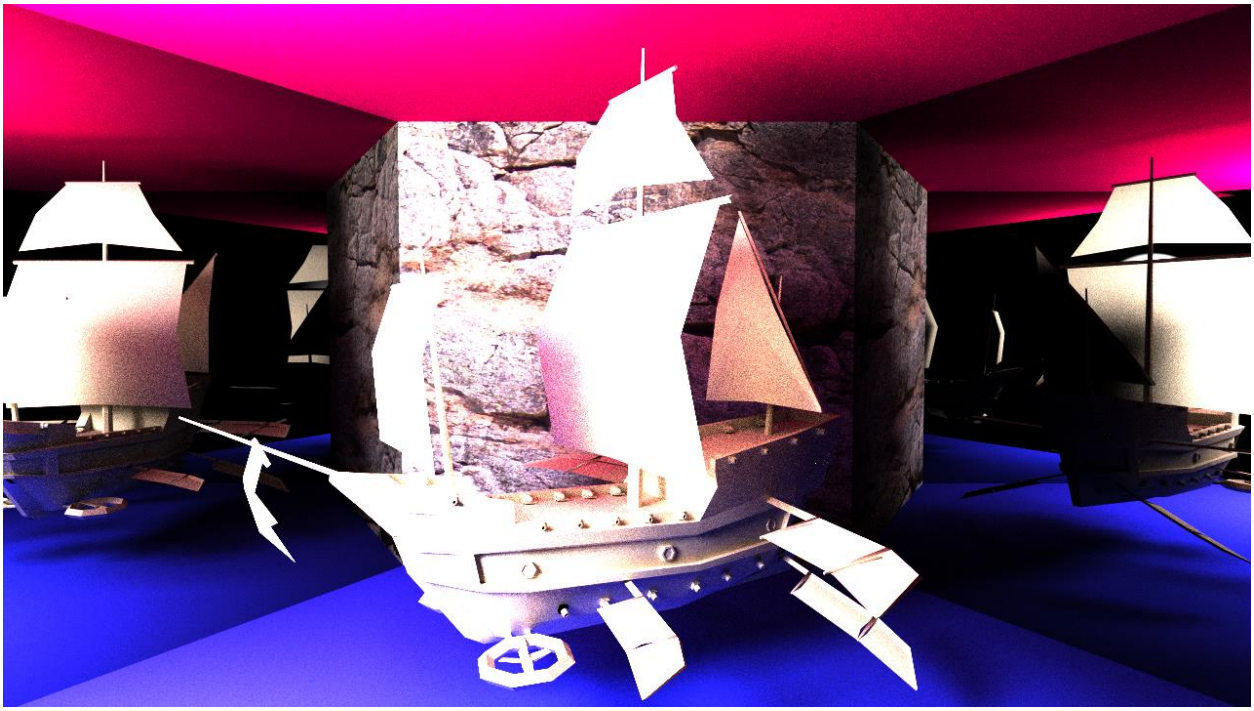
Please see below for details.

Scenes:

(4 scenes, 1 custom obj file and xml loader attributes for depth of field)

ship.obj is my homework from 3D modeling class.

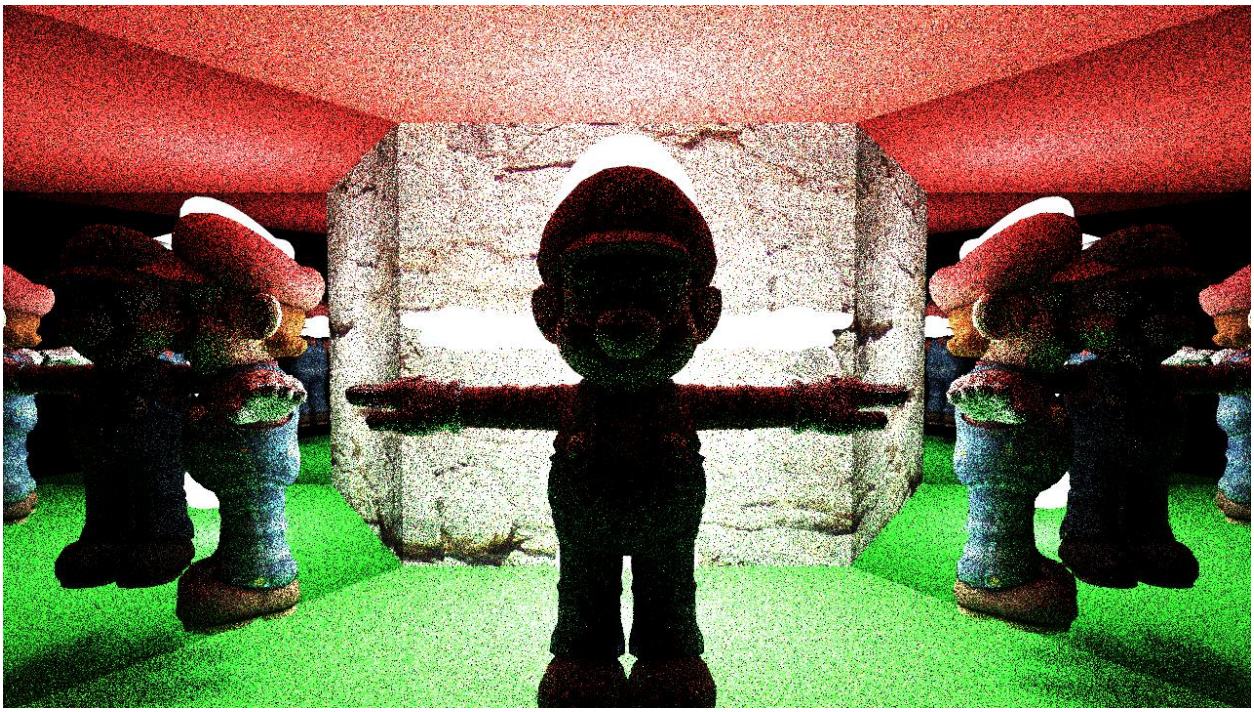
1. ship.xml:



2. dragon_hell.xml: six dragons!

Takes some time to construct my BVH when loading. Rendered in 38 seconds (1 sample per pixel, 1280x720) so at least my SAH is working.

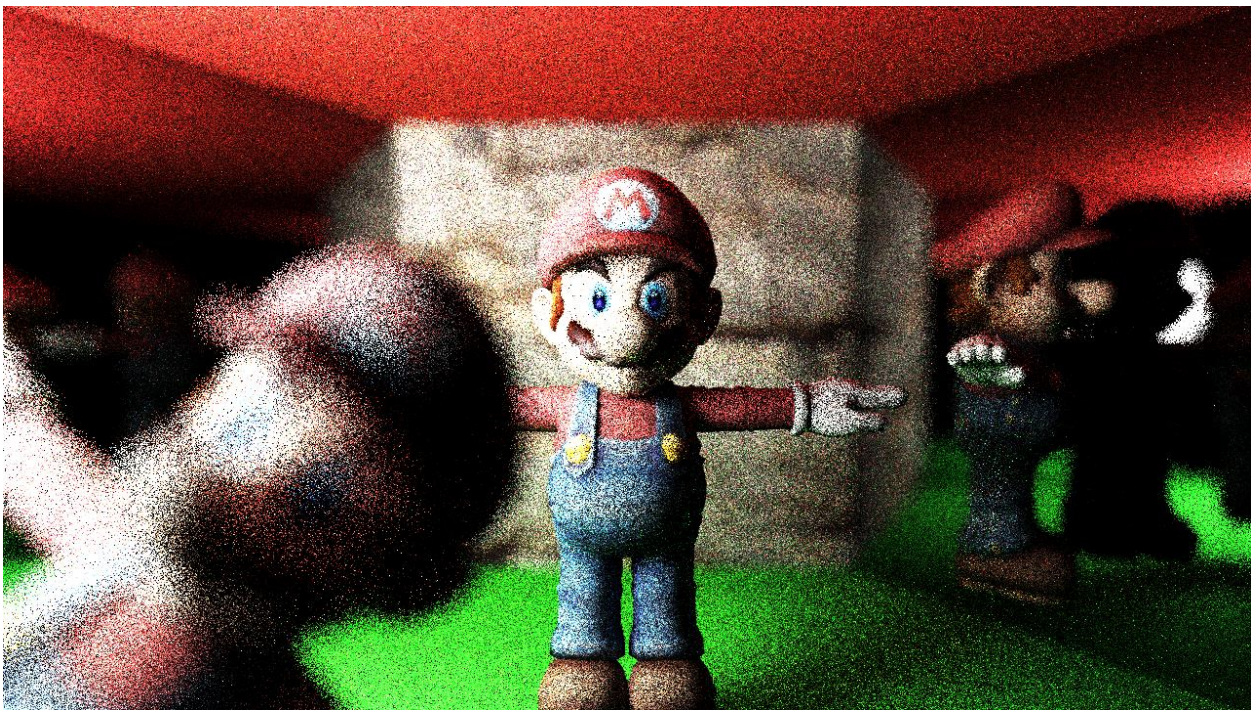
3. holy_wahoo.xml: BEHOLD! THE POWER OF THE MIGHTY ■■■■■ (one behind as light source)



4. focal_wahoo.xml: to test depth of field.

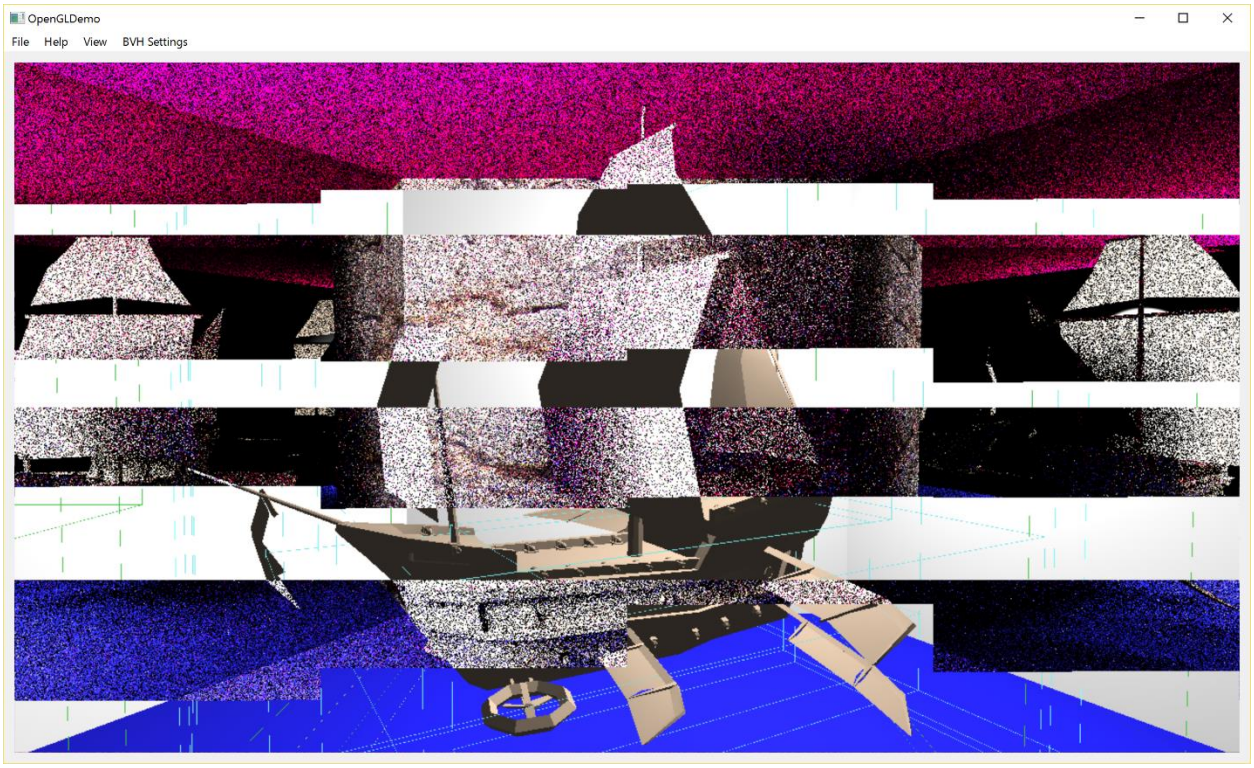
```
<FocalDistance>9</FocalDistance>
<LensRadius>0.25</LensRadius>
```

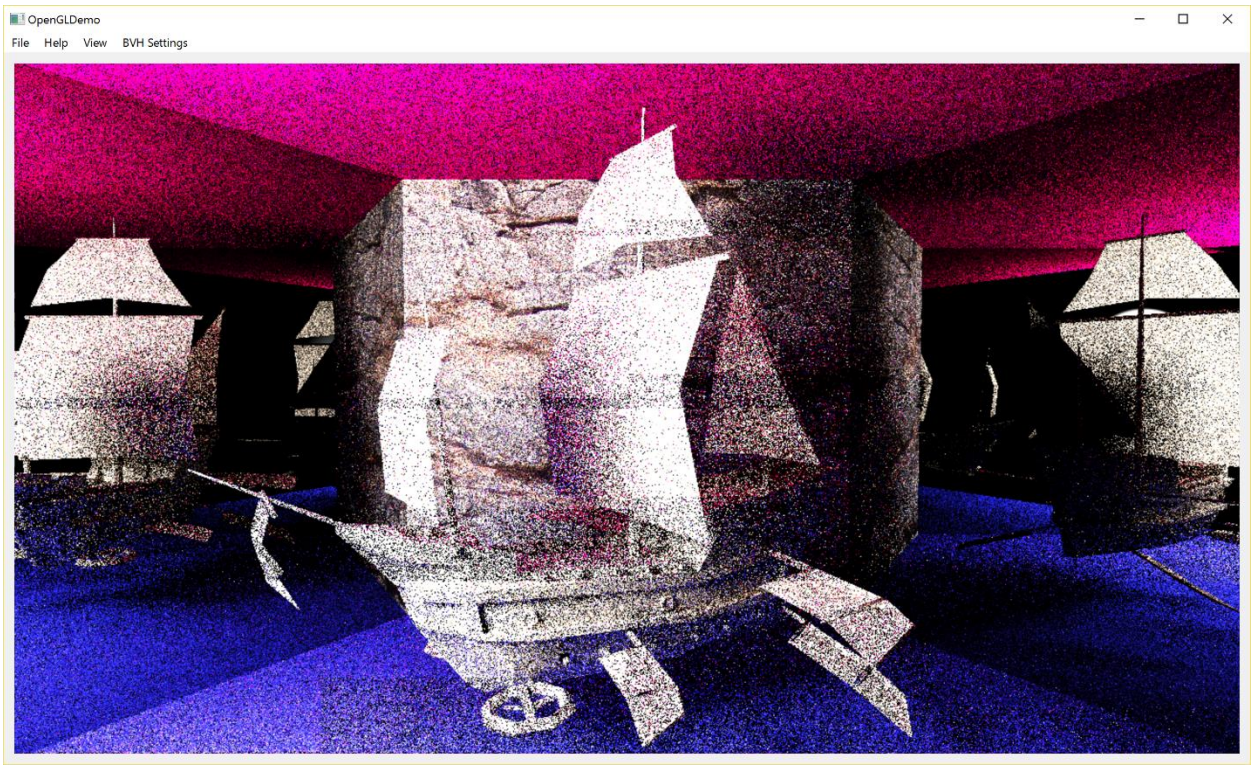
Attributes for depth of field

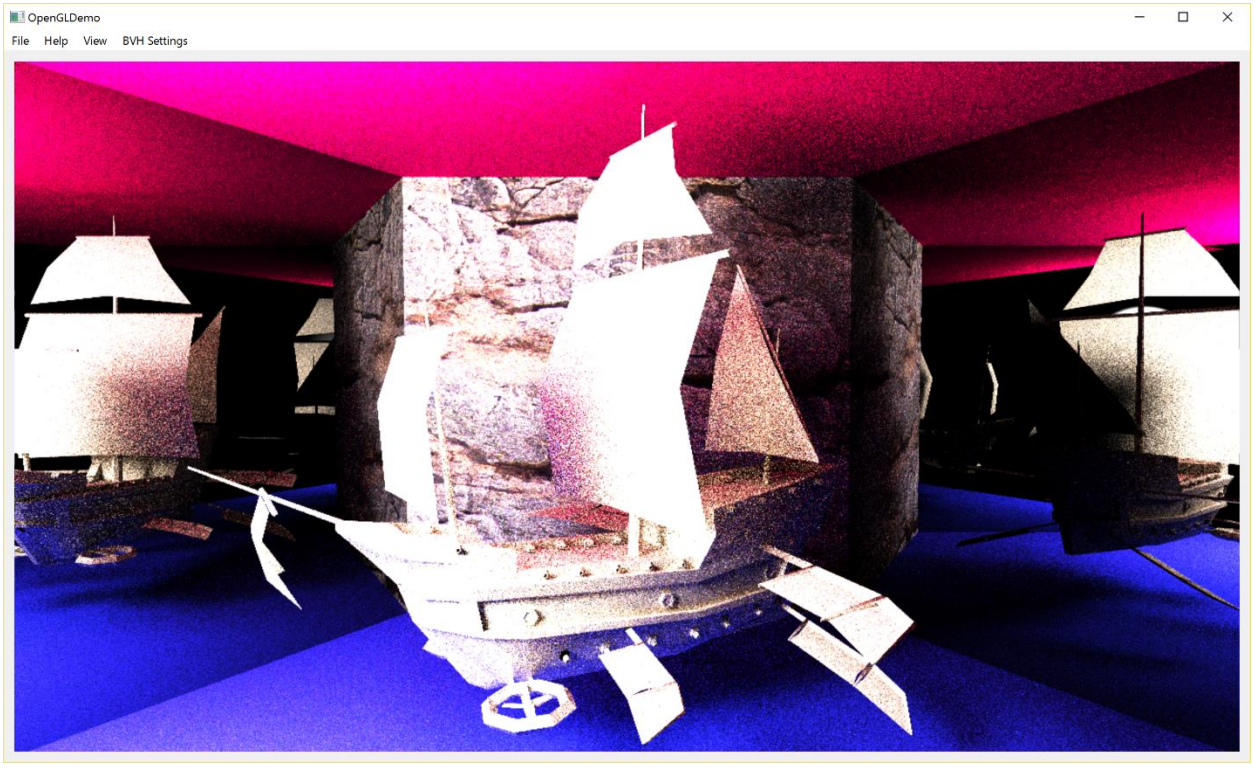


Progressive Rendering

I use OpenGL to draw a textured quad on the scene and modify the texture based on the rendering process. Also, I changed the loop order in RenderThread so it renders 1 sample for every pixel per pass and renders another sample in the second pass and so on.




OpenGLDemo
File Help View BVH Settings


OpenGLDemo
File Help View BVH Settings

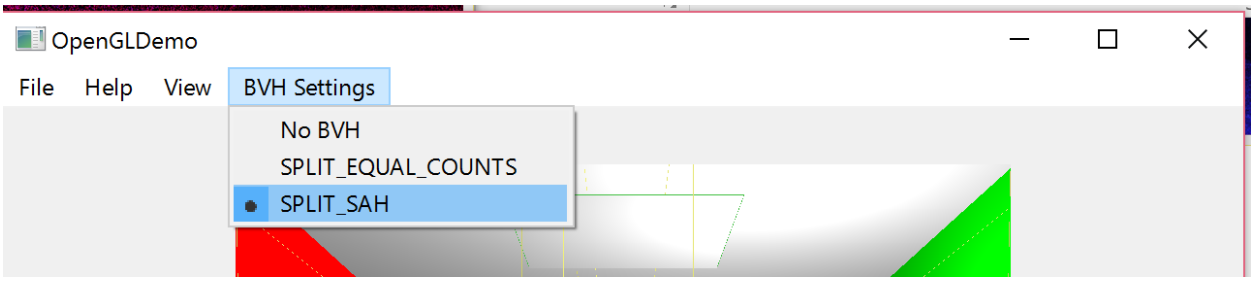
BVH integration and the surface area heuristic

I have implemented BVH integration and the surface area heuristic. I also allowed switching between BVH modes (need to reload the scene file)

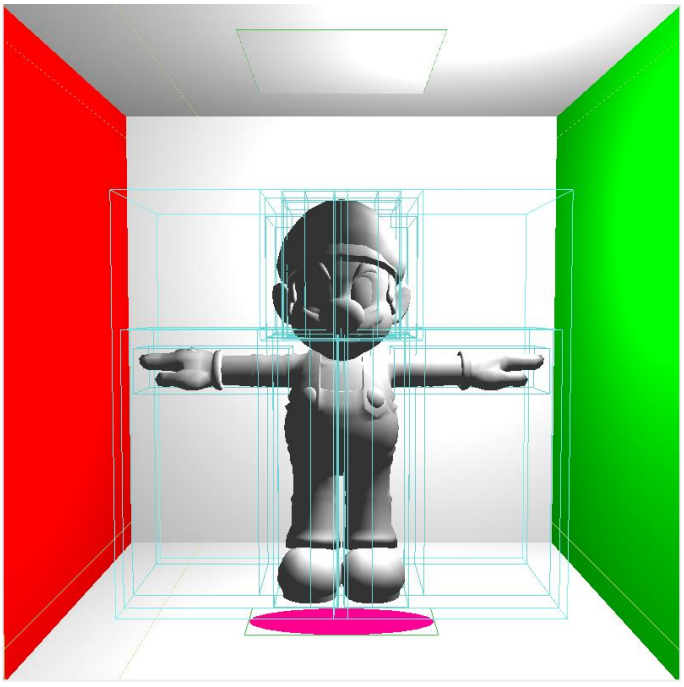
Render time for my dragon_hell.xml:

38.5s when using SPLIT_SAH (59.0s when using SPLIT_EQUAL_COUNTS)

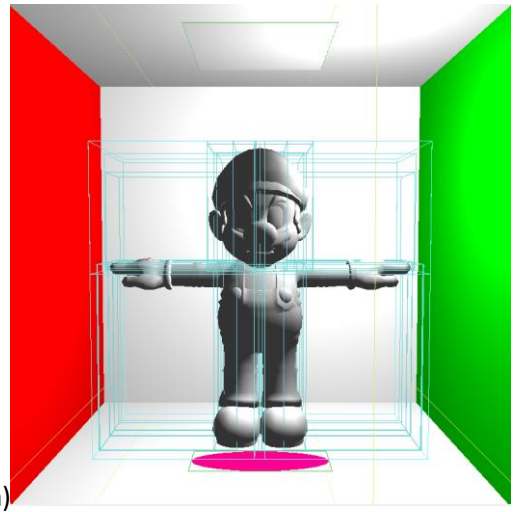
See raytracing/bvhnode.h (/.cpp)



Wahoo with SPLIT_SAH:



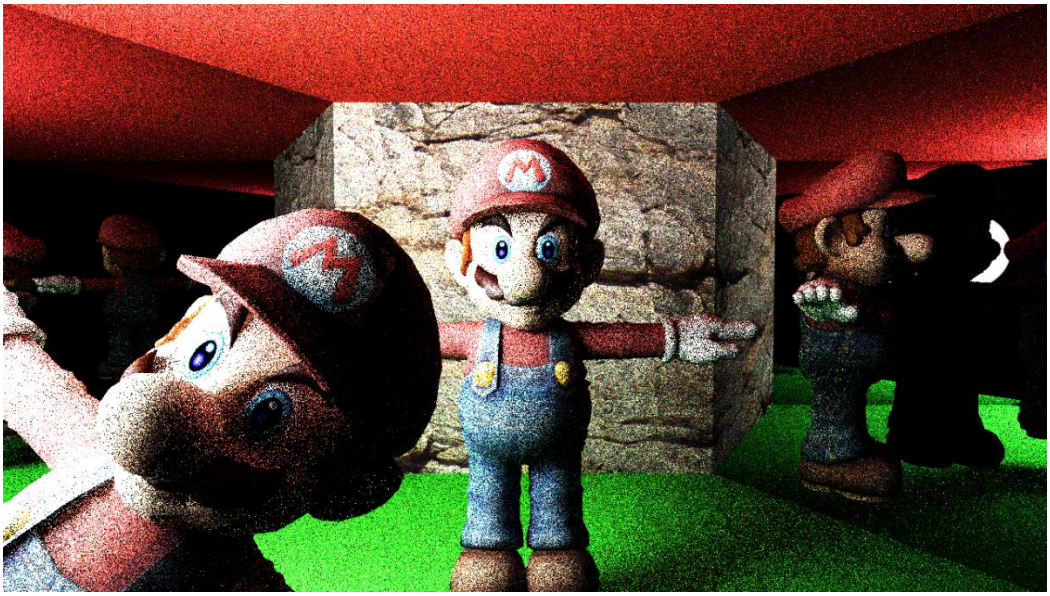
Wahoo with SPLIT_EQUAL_COUNTS (comparison)

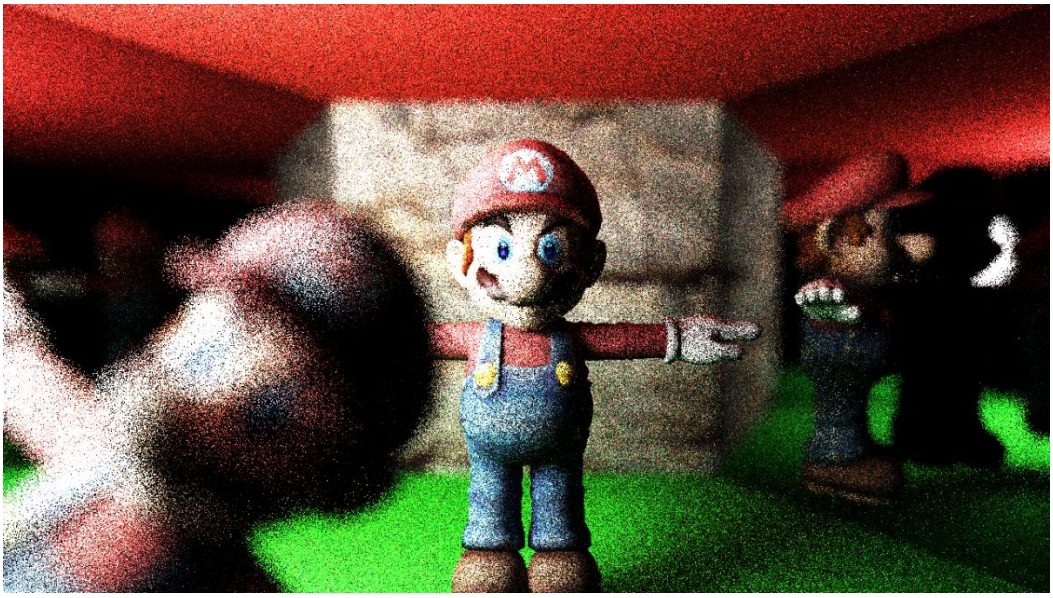


Depth of field

```
<FocalDistance>9</FocalDistance>
<LensRadius>0.25</LensRadius>
```

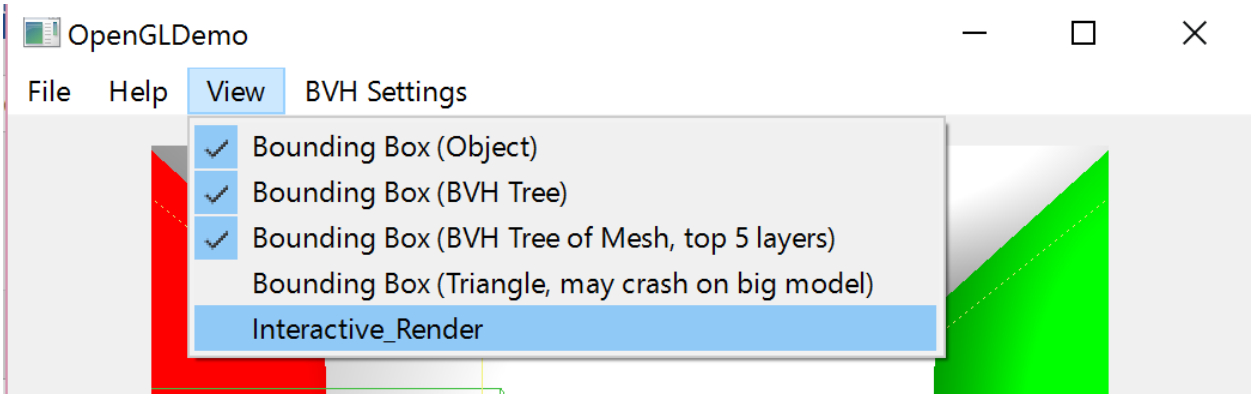
New attributes for XMLReader





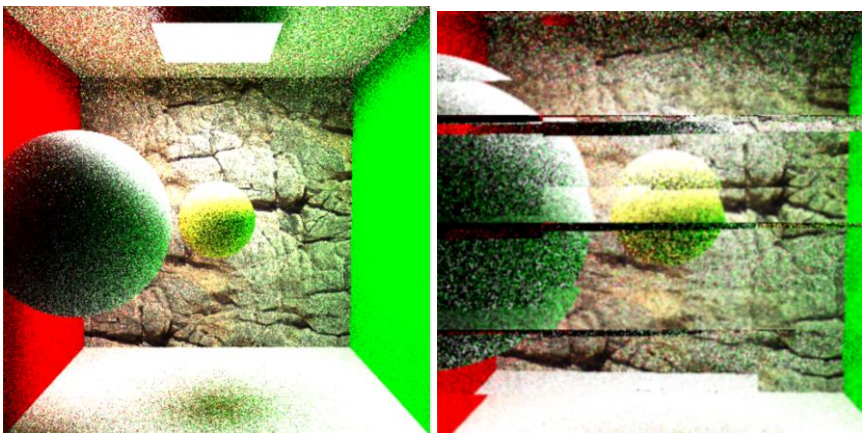
With and without depth of field

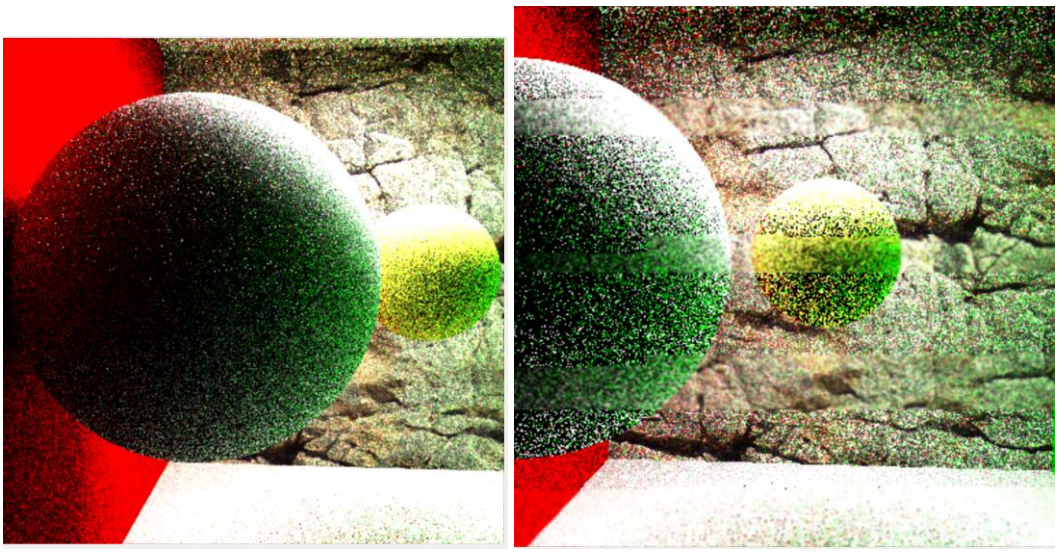
Other interesting features: Interactive Rendering



Use the View->Interactive Render to enable interactive rendering mode

In this mode, you can use the keyboard to navigate around while in rendering view! (I planned to do photon mapping to run this view much faster since I only need to do one prepass but I ran out of my time to finish photon mapping)





Navigating around using WSAD and arrow keys in rendering mode.