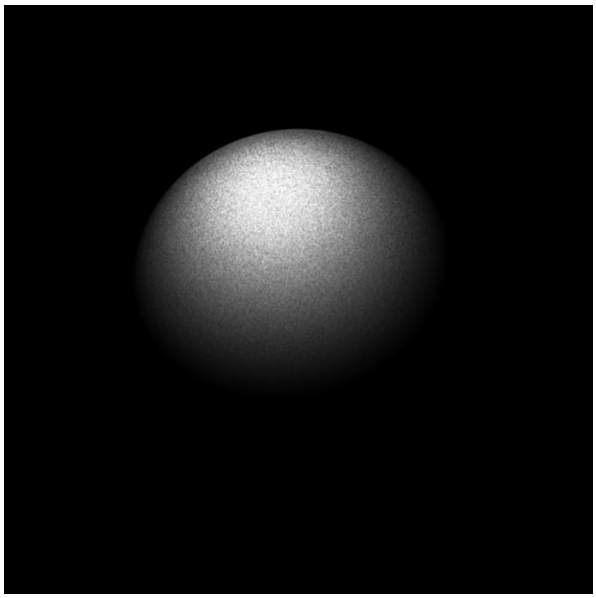
PROJECT 3

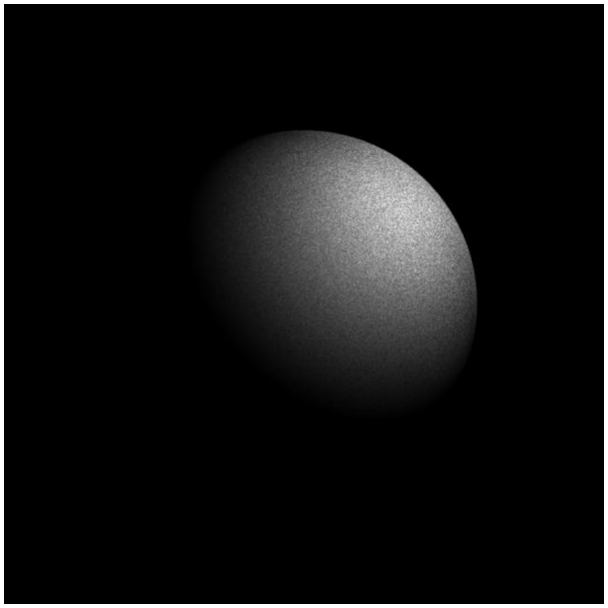
I Serial Version

To get the Grayscale Image, I multiplied the result by 255 to produce the following image, with the following arguments : n = 1000 and N_rays = 10000000.

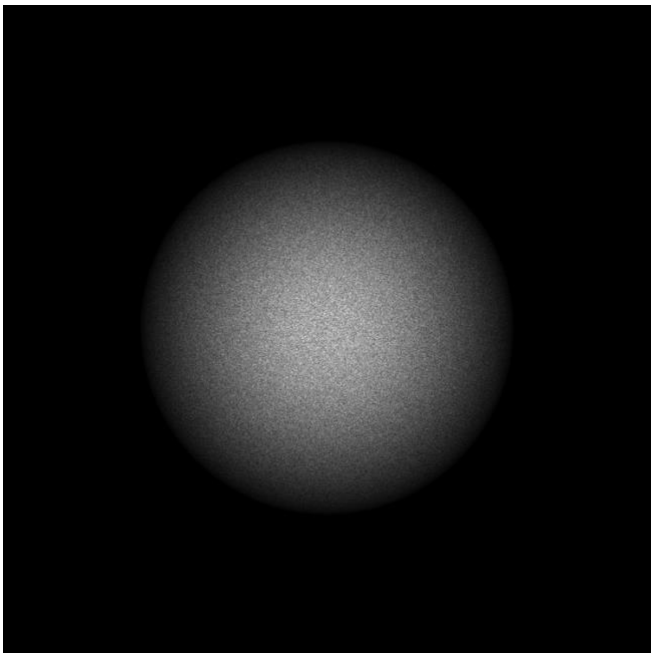
The sample image with L = (4, 4, -1), Wy = 10, Wmax = 10, C = (0, 12, 0), R = 6.
By setting n = 1000 and N_rays = 10000000, I got the following image. It runs in 74 seconds.



By only changing the position of light source to L = (4, 4, 4). I got the following imgae:



By only changing the position of light source to L = (0, 0, 0). I got the following imgae:



II Cuda Version

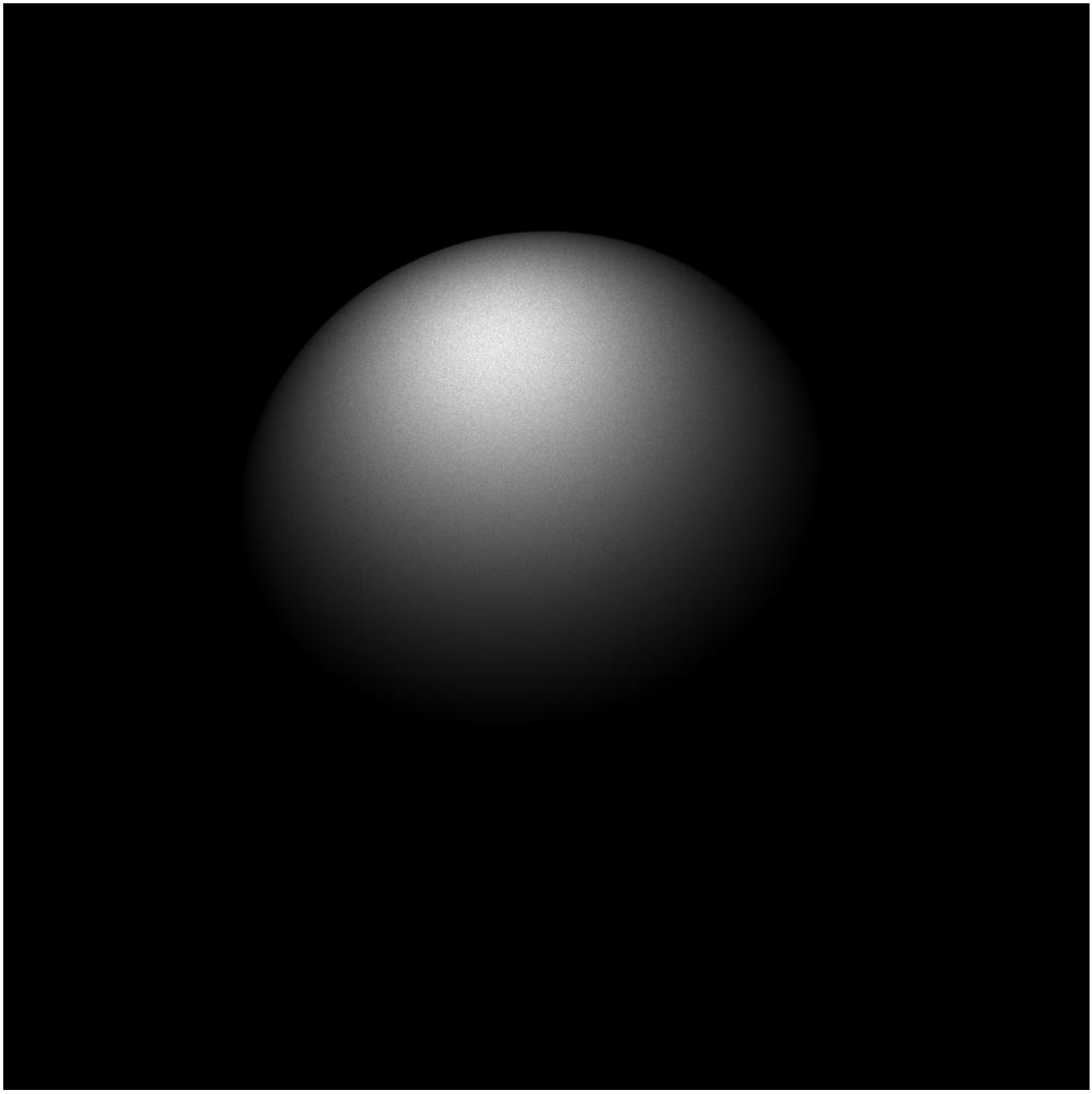
i Experiment with different block and thread configuration

Runs on Midway3, we can see that when Number of Threads Per Block = 8, have the shortest running time. And the max number of threads per block is 1024.

Number of Blocks	Number of Threads Per Block	Time(s)
10000	1024	2.39
20000	512	2.50
40000	256	2.58
80000	128	2.30
160000	64	2.23
320000	32	1.59
640000	16	1.28
1280000	8	1.17
2560000	4	1.18
5120000	2	1.31
10240000	1	1.31

ii Sample Image

Get the following picture with 1,000,000 blocks, and 1024 threads per block in 238.35s or with 125,000,000 blocks and 8 threads per block in 106.59s. Same as the table above, when the number of threads per block = 8, the performance is much better.



III Compare serial CPU version and CUDA GPU version

The problem scale: 10000000 Rays, grid size = 1000 * 1000.
CPU: Runs on miway3
GPU configuration: 1280000 blocks, 8 thread per blocks. Runs on midway3.

CPU Running Time(s)	GPU Running Time(s)
74	1.17

We can see that GPU version is much faster than CPU version, because not only we have a strong GPU but also GPU are design for specific task like this one, it has many small processor that can do a large number of small task at same time.