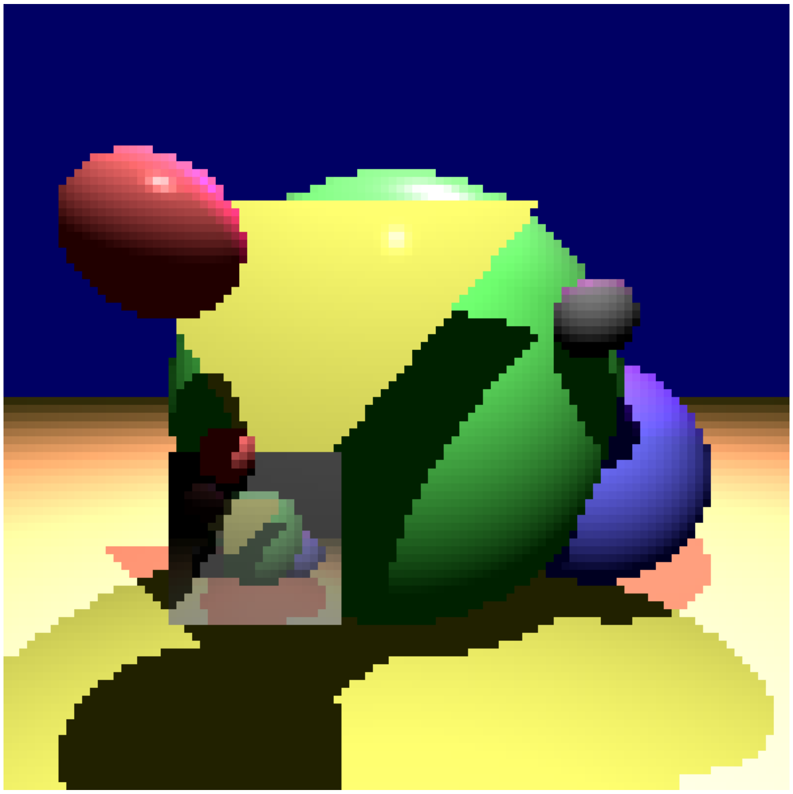
Alex Aulabaugh(aulabau2)

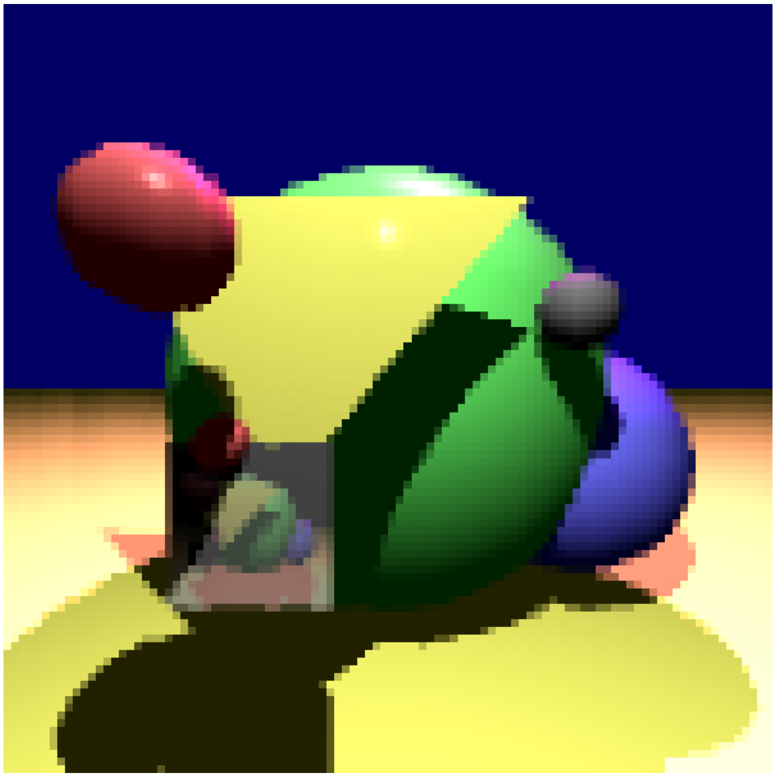
Ray Tracer Manual Test Plan

1. Anti-Aliasing Testing
   1. Navigate to “Final Project”/src/rendering/TracingCoordinator.java
   2. Edit line 434 in the constructor to read “buildSettings(“src/config/settings_Aliasing.txt”);"
   3. Run TracingCoordinator.

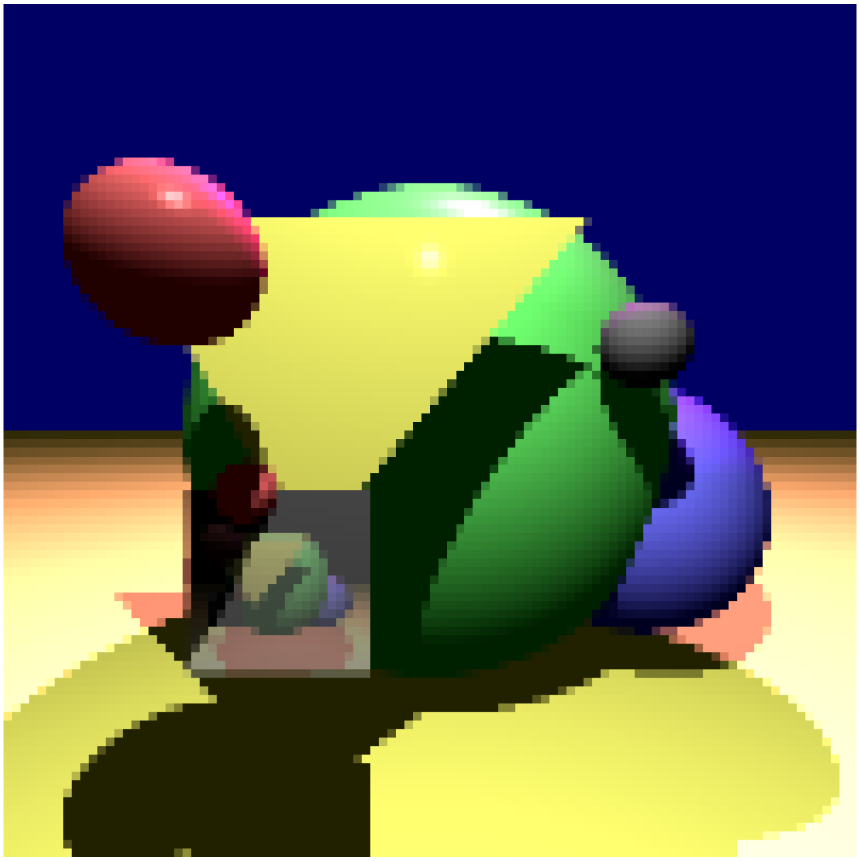


   4. This is the result of no sampling. Notice how jagged the edges are.
   5. Navigate to “FinalProject”/src/config/camera_Aliasing.txt
   6. Edit lines 9 and 10 to read
      SampleType:Random
      SampleSize:4
   7. Run TracingCoordinator

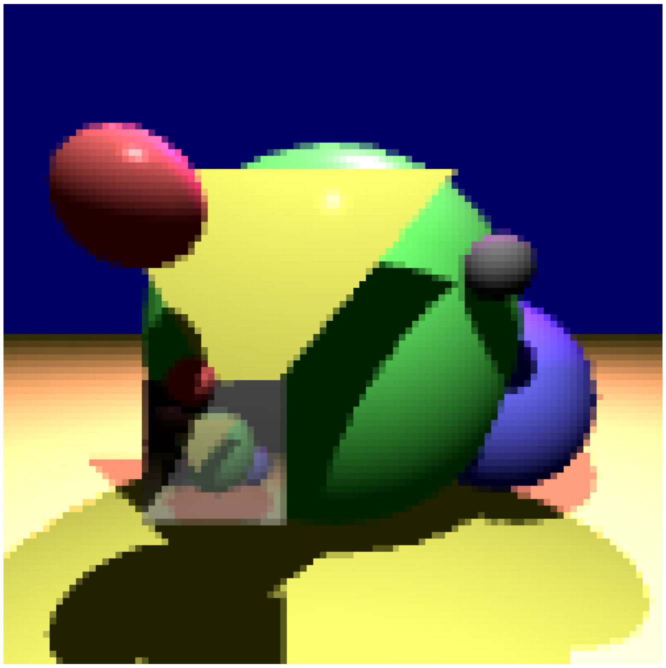
Alex Aulabaugh(aulabau2)



8. This is better, but on the red sphere in particular there are some weird effects going on. Also at the bottom of the rectangle.
9. Edit lines 9 and 10 of “FinalProject”/src/config/camera_Aliasing.txt to read
   SampleType:Uniform
   SampleSize:4
10. Run TracingCoordinator

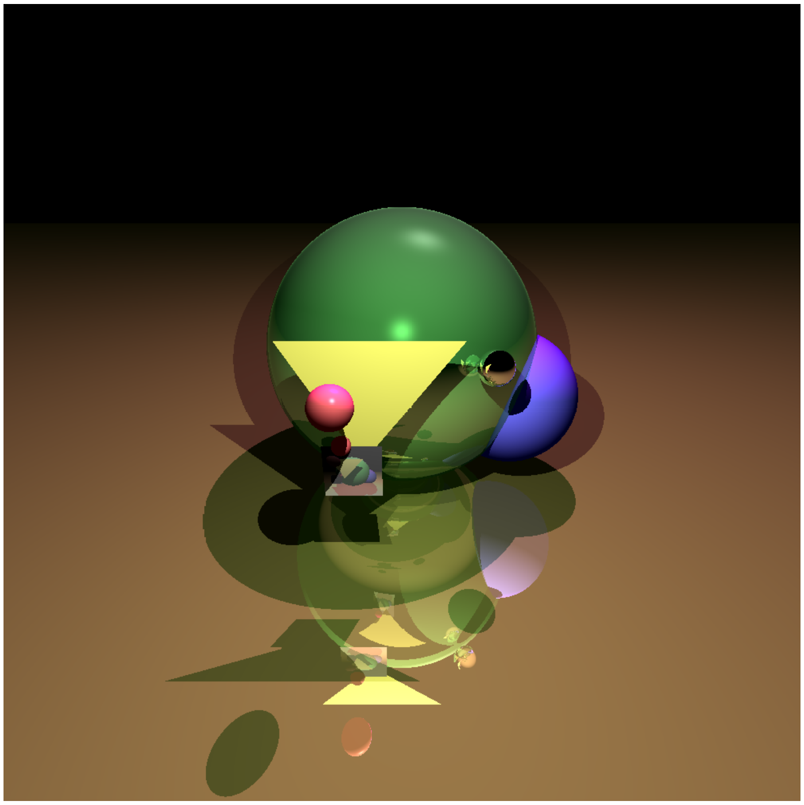


    11. Substantially better, but notice how the top and right of the rectangle are sharp while the bottom and left are blurred? This is an artifact of doing things without any randomness.
    12. Edit lines 9 and 10 of “FinalProject”/src/config/camera_Aliasing.txt to read
        SampleType:MultiJitter
        SampleSize:4



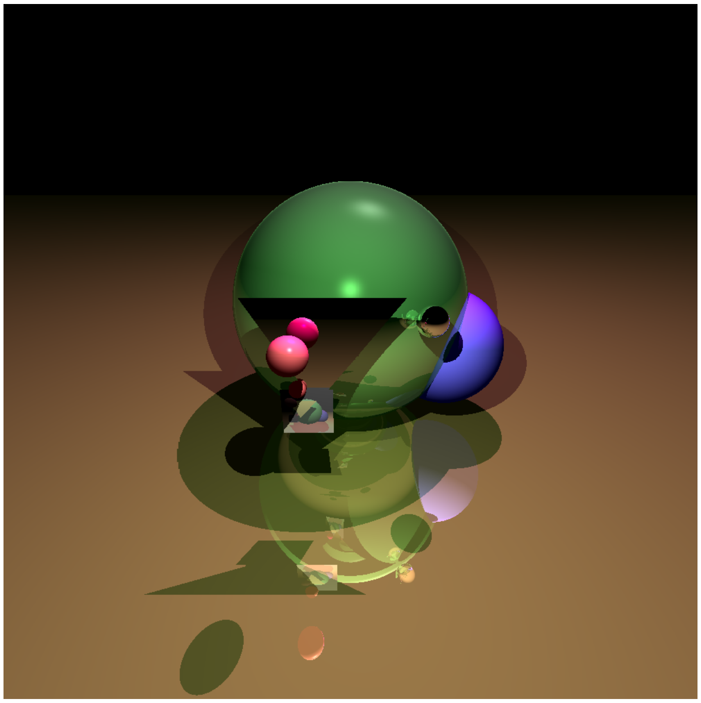
    13. This looks pretty good! It’s the most balanced of all we’ve observed, and plays nicely with repeating patterns in textures.
    14. Attempt to change the lines so that the number being sampled is not a perfect square, and to a nonexistent style. The program will warn then continue with no sampling.

2. Reflection
    1. Navigate to “Final Project”/src/rendering/TracingCoordinator.java
    2. Edit line 434 in the constructor to read
       “buildSettings(“src/config/settings_Reflect.txt”);”

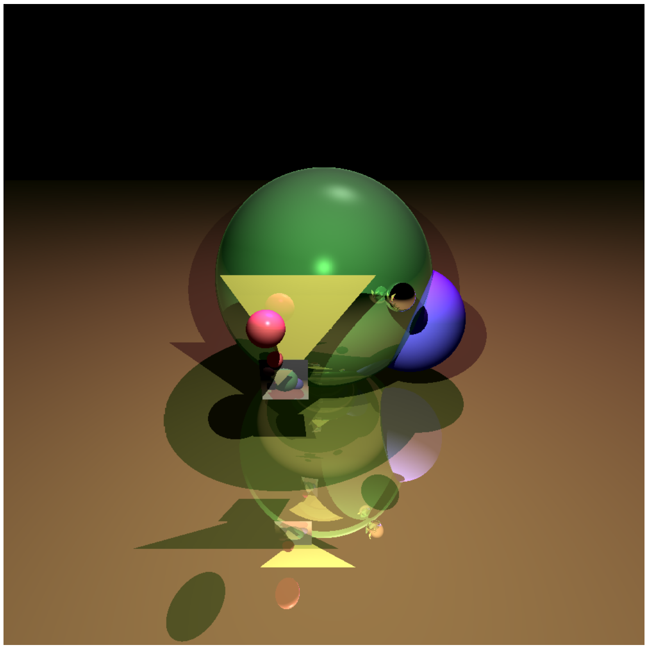


3. Here multiple iterations of reflection can be observed. Additionally, it can be seen that the small sphere on the right is much more reflective than the green large sphere. This can be controlled.
4. In “Final Project”/src/config/scene_Reflect.txt edit line 4 to “Triangle:-60,40,-80, 30,40,-80 -20,-20,-70 255,255,0 1250.0 1.0”
5. The last space separated param is the reflection fraction. We’ve just made this triangle a perfect mirror.
6. Run TracingCoordinator

Alex Aulabaugh(aulabau2)

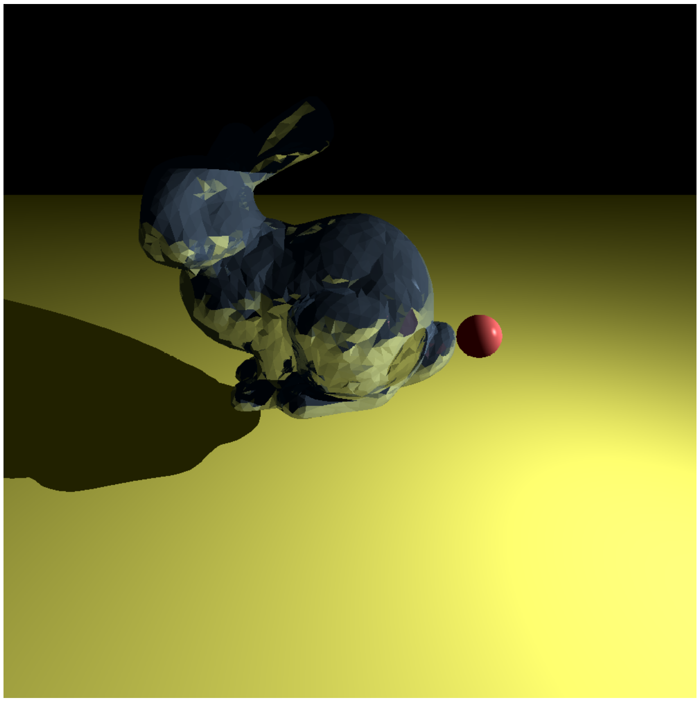


7. Now, let’s put it somewhere more moderate.
8. In “Final Project”/src/config/scene_Reflect.txt edit line 4 to “Triangle:-60,40,-80, 30,40,-80 -20,-20,-70 255,255,0 1250.0 .2”

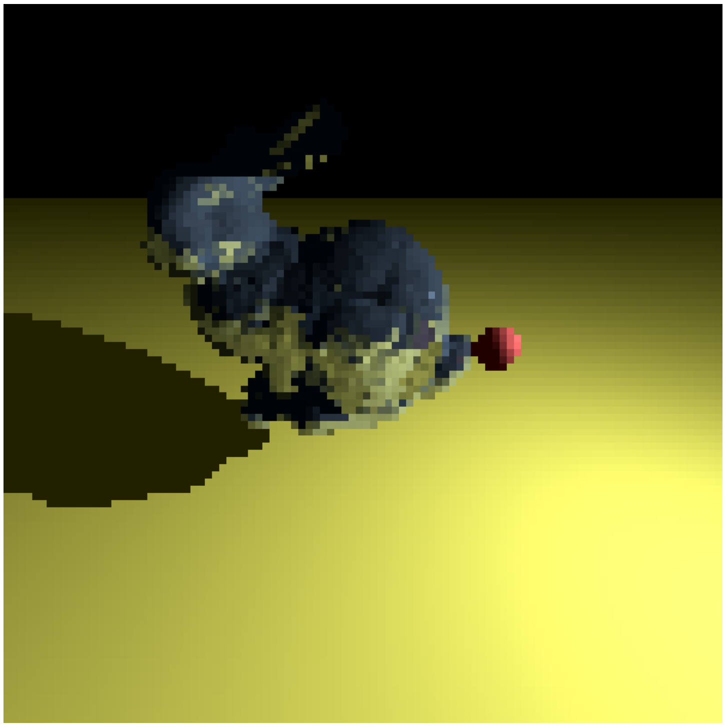


   9. Notice that most of the base color remains.

3. Triangle Meshes & Uniform Grid
   1. Navigate to “Final Project”/src/rendering/TracingCoordinator.java
   2. Edit line 434 in the constructor to read
      buildSettings(“src/config/settings_Bunny.txt")
   3. This reads in a bunny .obj file. Run TracingCoordinator
   4. It should run sub 20 seconds (Acceleration structure is ON)

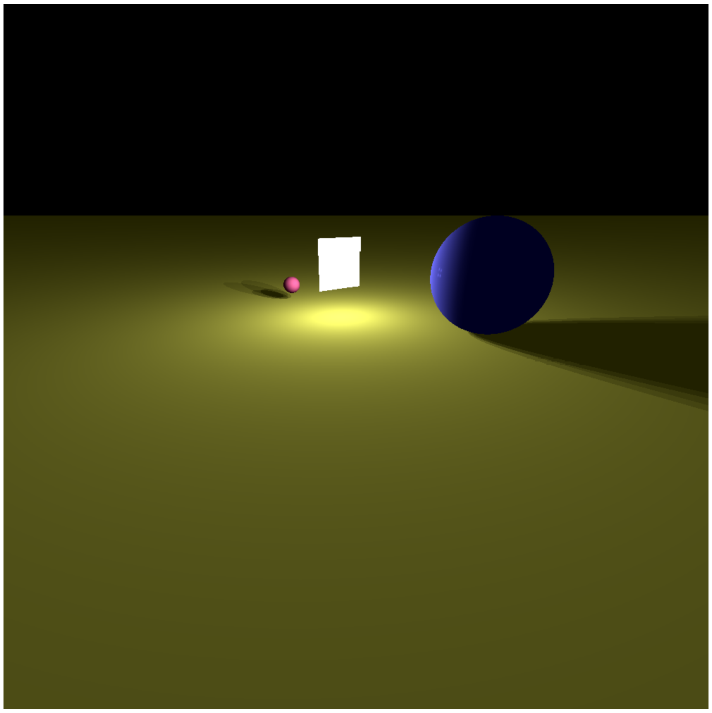


   5. Notice that not only is the bunny present and highly detailed (4969 Objects in the scene), but it has reflections. Acceleration structure are the best.
   6. To demonstrate that they’re the best, open config/settings_Bunny.txt and comment out
      RegularGrid:3
      to
      //RegularGrid:3
   7. Run again. Five minutes and no end in sight, I’d abort the process.
   8. Edit config/camera_Bunny.txt lines 6 and 7 to read
      wres:10
      hres:100
      Run again

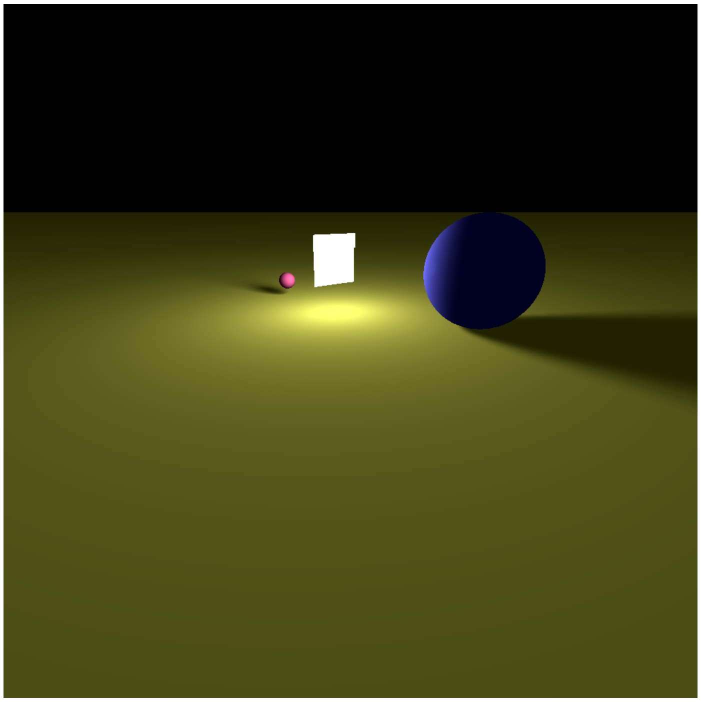


   9. Should succeed in < 1 min, but the quality is poor.

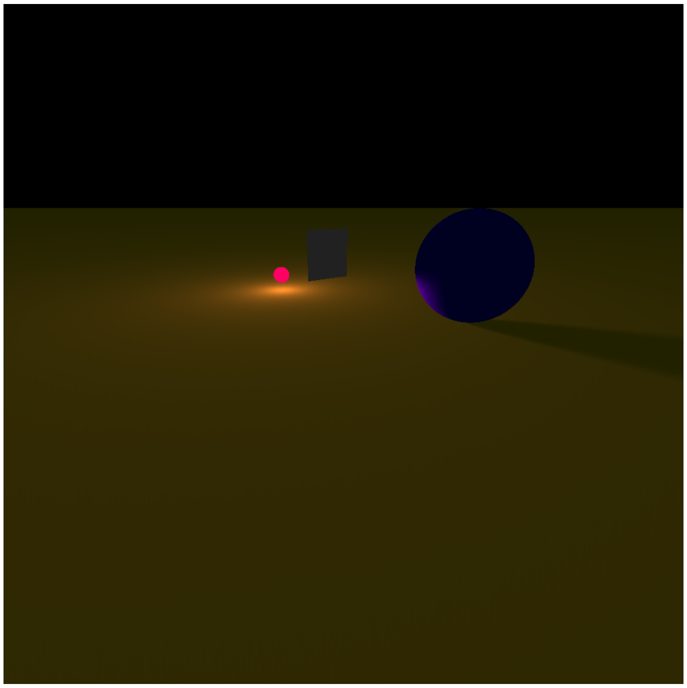
4. AreaLights
   1. Navigate to “Final Project”/src/rendering/TracingCoordinator.java
   2. Edit line 434 in the constructor to read buildSettings(“src/config/settings_AreaLight.txt”)
   3. Run



4. If you look closely you an see bands in the shadows, this is because we're using uniform sampling.
5. Edit scene_AreaLight.txt lines 2 and 3
   EmitterSampleType:Uniform
   EmitterSampleSize:36

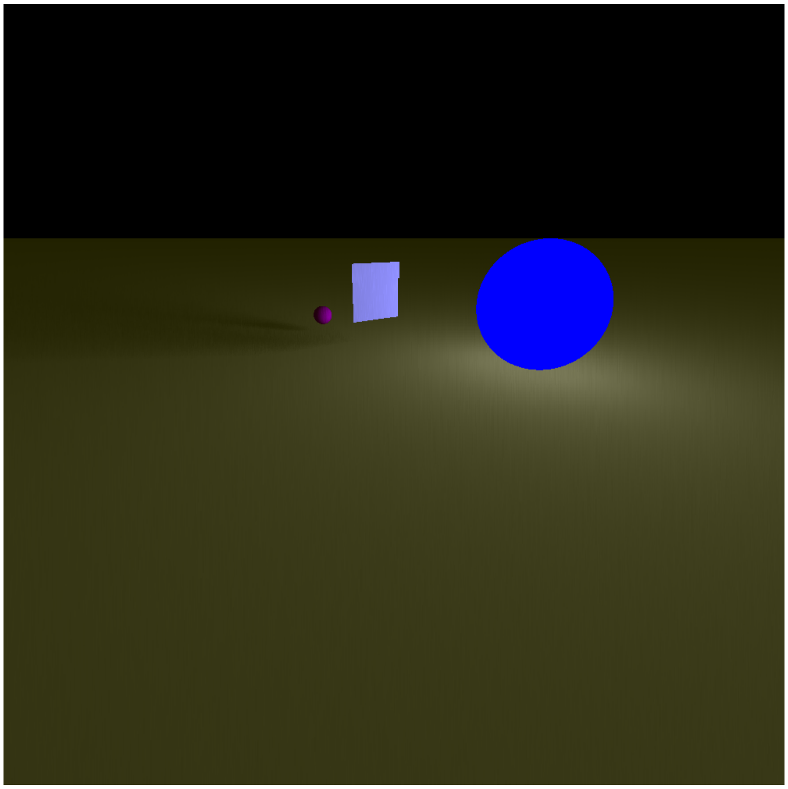


6. This helps a lot, the penumbras are very smooth now. Try Random & MultiJittering also for sampling. Though they should be much more noisy. (especially random)
7. Edit lines 4,5,6 to read
   Sphere:-50,-80,-400 10.0 255,0,100 none 1250.0 0.0 True
   Sphere:30,-50,-180 50.0 0,0,255 none 1250.0 0.0 False
   Rectangle:-70,-70,-300 0,50,0 50,0,0 255,255,255 none 1250.0 0.0 False



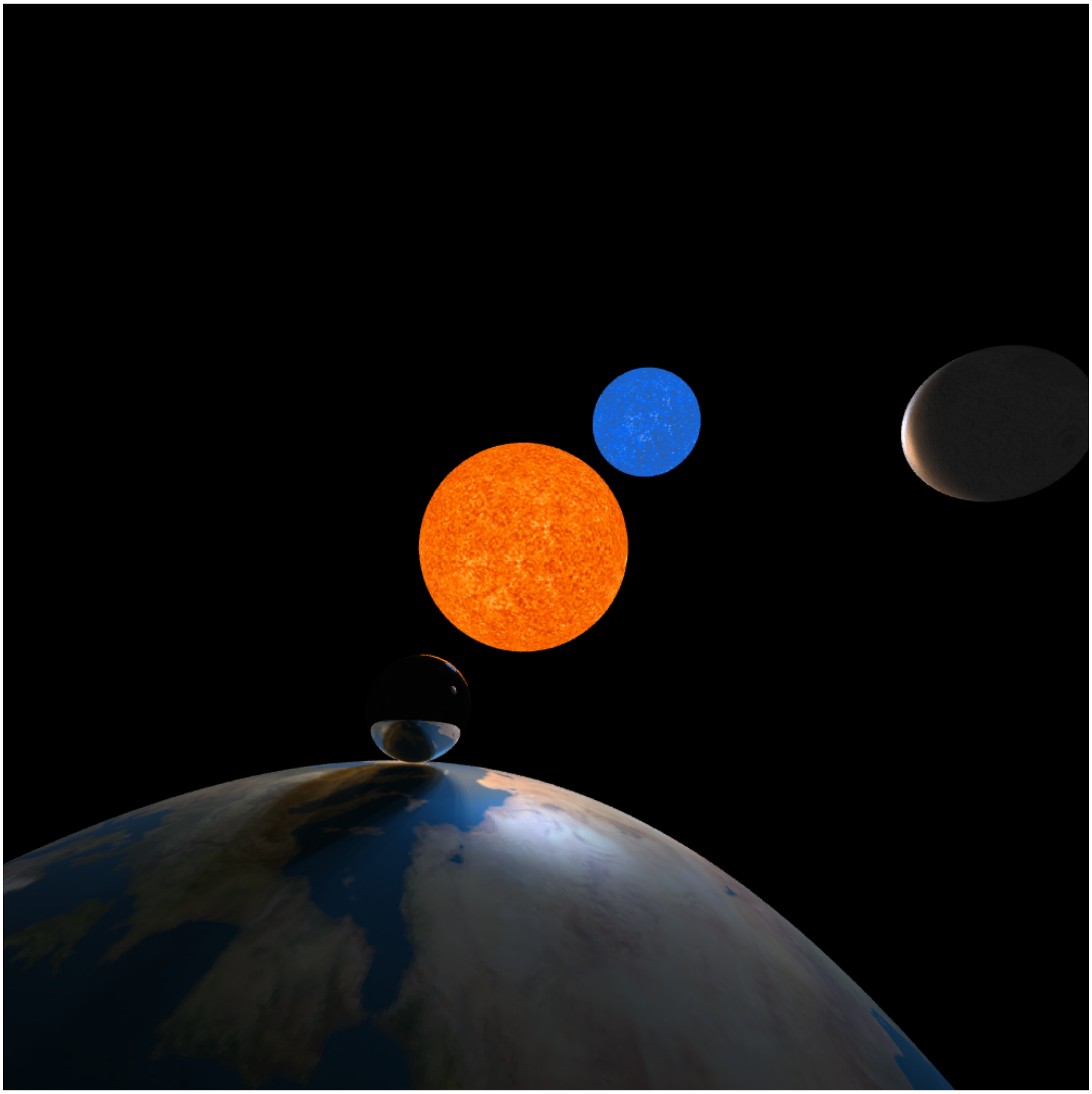
8. And likewise
   Sphere:-50,-80,-400 10.0 255,0,100 none 1250.0 0.0 False
   Sphere:30,-50,-180 50.0 0,0,255 none 1250.0 0.0 True
   Rectangle:-70,-70,-300 0,50,0 50,0,0 255,255,255 none 1250.0 0.0 False

   These falses/trues enable/disable the object from being an area light.



9. We can also make spheres and rectangles into area lights when they have images on them. This is demonstrated next.

5. Sphere Images
   1. Navigate to “Final Project”/src/rendering/TracingCoordinator.java
   2. Edit line 434 in the constructor to read “buildSettings(“src/config/settings_SolarSystem.txt”);"
   3. Run TracingCoordinator.



(SolarSystem.png, not testOut.png)

4. This demonstrates images on spheres with lighting, as well as area lights with images on them. You can see the distance red and blue components of light on the Moon.