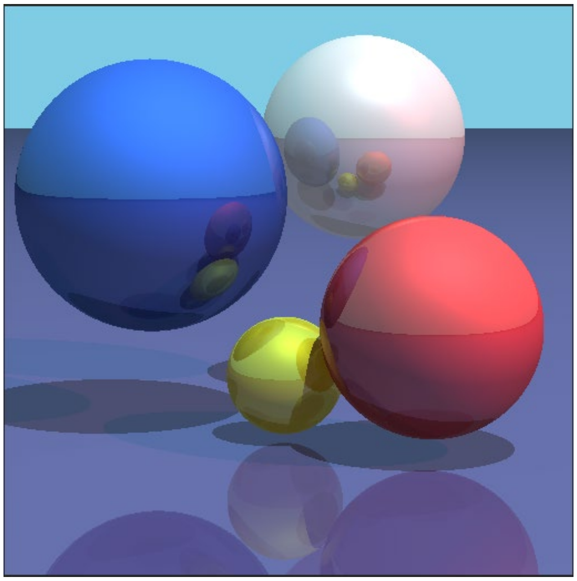
Kyle Andrus
914056131

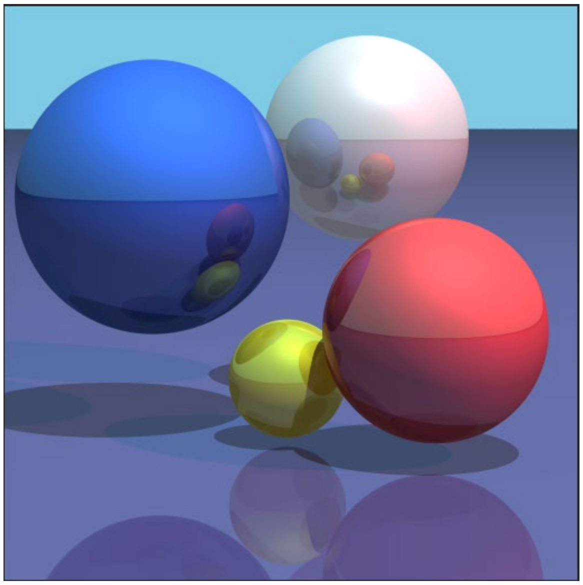
ECS 275A Program 2

The base image shows the final ray traced product of Program 1 without the effects of anti-aliasing, depth of field and motion blur. The maximum ray depth is 4.



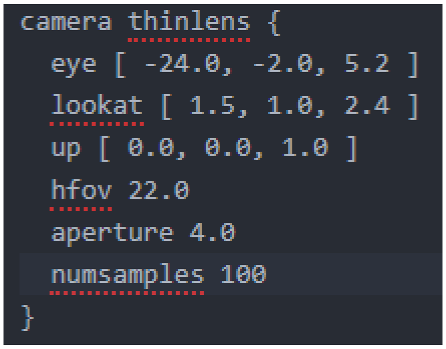
Anti-Aliasing

In the input file, I specify the number of samples to take per pixel labelled ‘samplesperpixel’. I found that 25 samples using jittered sampling produce a good enough effect, so the below image displays that result. The end effect shows smoothness around the edge of the spheres and the edge of the plane.

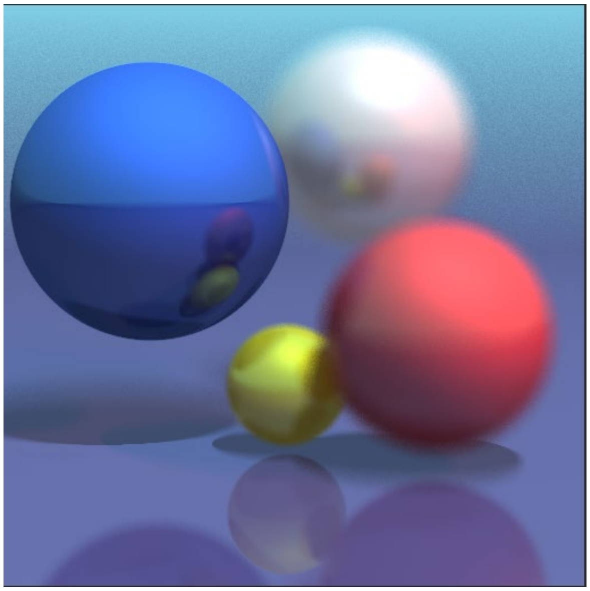


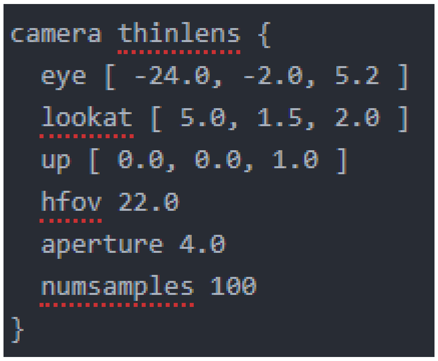
Depth of Field

In the input file, I specify the parameters for a thin lens camera to showcase a depth of field effect. It keeps the same 4 parameters as the pinhole camera (eye, lookat, up, and hfov), but I add the aperture value and the number of samples to use. The focal length is computed from the length of the vector (lookat – eye). Due to the random sampling, the resulting image is quite noisy, so I increased the number of samples to 100 to substantially reduce this. The end effect shows one sphere in focus while the others are noticeably out of focus.

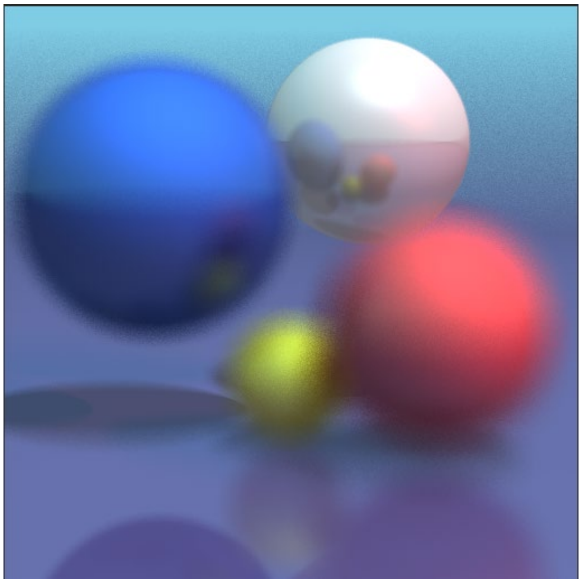


```
camera thinlens {
  eye [ -24.0, -2.0, 5.2 ]
  lookat [ 1.5, 1.0, 2.4 ]
  up [ 0.0, 0.0, 1.0 ]
  hfov 22.0
  aperture 4.0
  numsamples 100
}
```



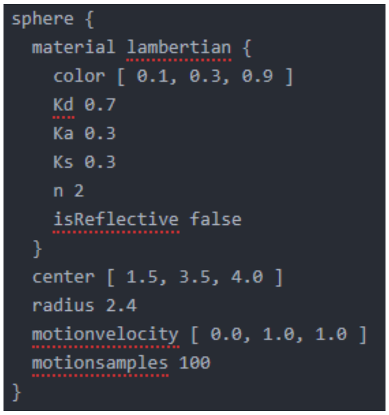


```
camera thinlens {
  eye [ -24.0, -2.0, 5.2 ]
  lookat [ 5.0, 1.5, 2.0 ]
  up [ 0.0, 0.0, 1.0 ]
  hfov 22.0
  aperture 4.0
  numsamples 100
}
```

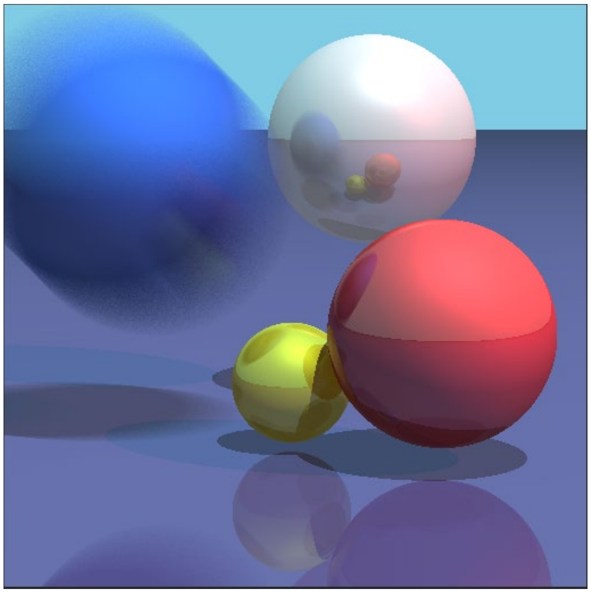


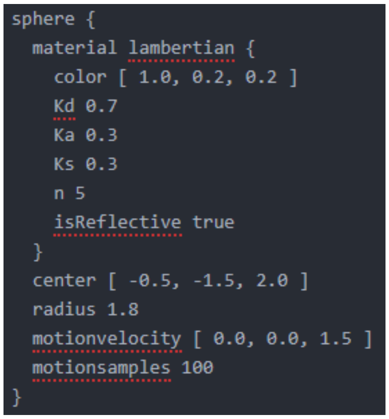
Motion Blur

In the input file, I add to a primitive's parameters by specifying the motion velocity and the number of samples to use. To create the motion effect, I multiply the velocity vector by a random t value and add that to the center of the primitive, in this case the sphere. Again, due to the random sampling, the resulting image is quite noisy, so I increased the number of samples to 100 to substantially reduce this. The end effect shows one sphere seemingly moving in the image.

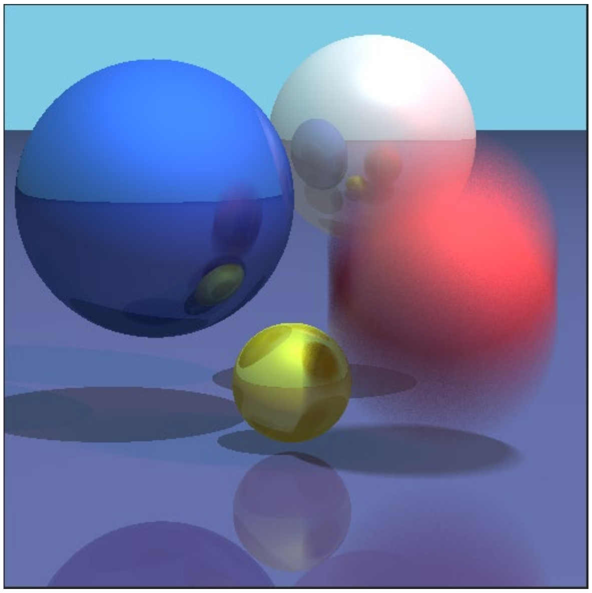


```
sphere {
  material lambertian {
    color [ 0.1, 0.3, 0.9 ]
    Kd 0.7
    Ka 0.3
    Ks 0.3
    n 2
    isReflective false
  }
  center [ 1.5, 3.5, 4.0 ]
  radius 2.4
  motionvelocity [ 0.0, 1.0, 1.0 ]
  motionsamples 100
}
```



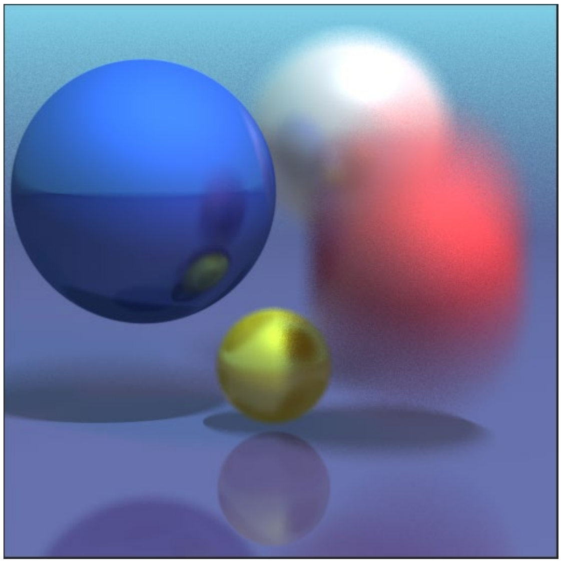


```
sphere {
  material lambertian {
    color [ 1.0, 0.2, 0.2 ]
    Kd 0.7
    Ka 0.3
    Ks 0.3
    n 5
    isReflective true
  }
  center [ -0.5, -1.5, 2.0 ]
  radius 1.8
  motionvelocity [ 0.0, 0.0, 1.5 ]
  motionsamples 100
}
```

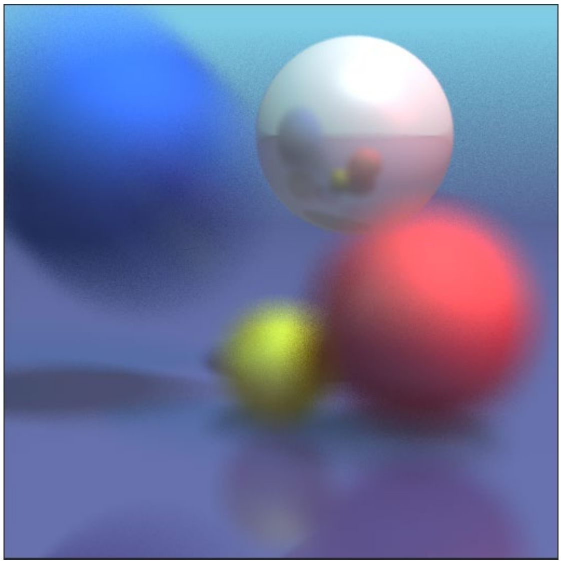


Combined Effects

Anti-aliasing, depth of field, and motion blur are all active in the following images. Even though I specify a separate "number of samples" parameter for each effect, the anti-aliasing parameter is used for depth of field and motion blur when used in conjunction with anti-aliasing.



Blue sphere in focus, red sphere in motion



White sphere in focus, blue sphere in motion

Even though the program description didn't specify to submit source files, I've submitted Scene.cc and ThinLensCamera.cc just in case. I've made small changes in several files throughout the program, but these two showcase the most work to produce distributed ray tracing.