**Rendering - HW2**

Result:

1. dof-dragons.dgauss.pbrt
2. 32 samples

C:\Users\sc\Documents\pbrt-v2-master\hw2\a.tiff

1. 512 samples

C:\Users\sc\Documents\pbrt-v2-master\hw2\a1.tiff

1. dof-dragons.fisheye.pbrt
2. 32 samples

C:\Users\sc\Documents\pbrt-v2-master\hw2\c.tiff

1. 512 samples

C:\Users\sc\Documents\pbrt-v2-master\hw2\c1.tiff

1. dof-dragons.telephoto.pbrt
2. 32 samples

C:\Users\sc\Documents\pbrt-v2-master\hw2\b.tiff

1. 512 samples

C:\Users\sc\Documents\pbrt-v2-master\hw2\b1.tiff

1. dof-dragons.wide.pbrt
2. 32 samples

C:\Users\sc\Documents\pbrt-v2-master\hw2\d.tiff

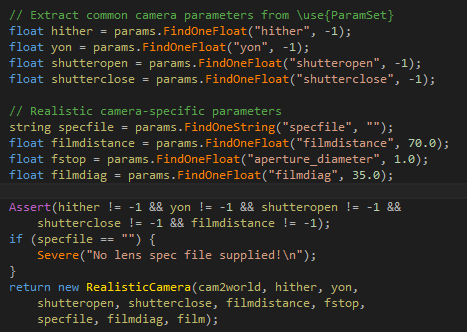
1. 512 samples

C:\Users\sc\Documents\pbrt-v2-master\hw2\d2.tiff

Method:

1. Collect relation data – initial()
   1. Read XXXX.pbrt data

Get by CreateRealisticCamera() method for system arthitecture.

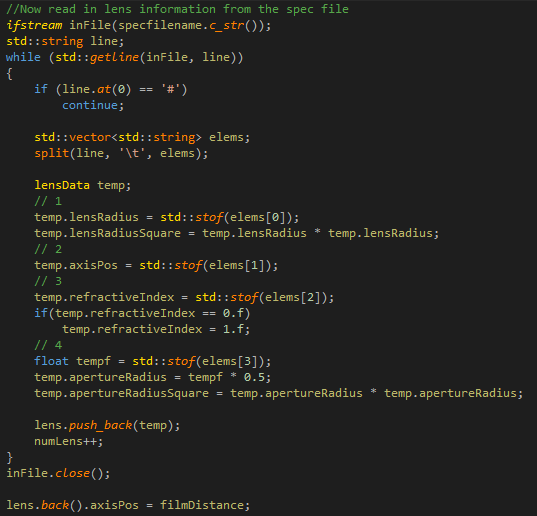


* 1. Read XXXX.dm file

In initial method - RealisticCamera()

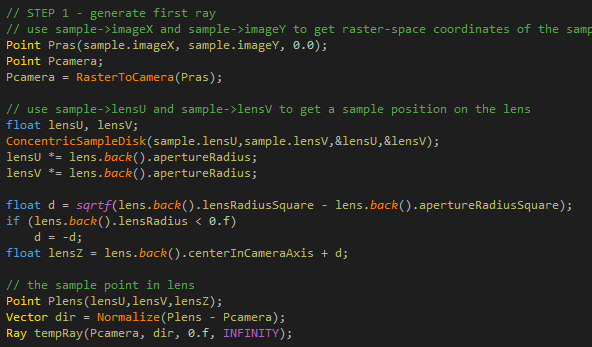
Read file and split by ‘tab’ to get float data.

All store in a vector by lensData data structure.



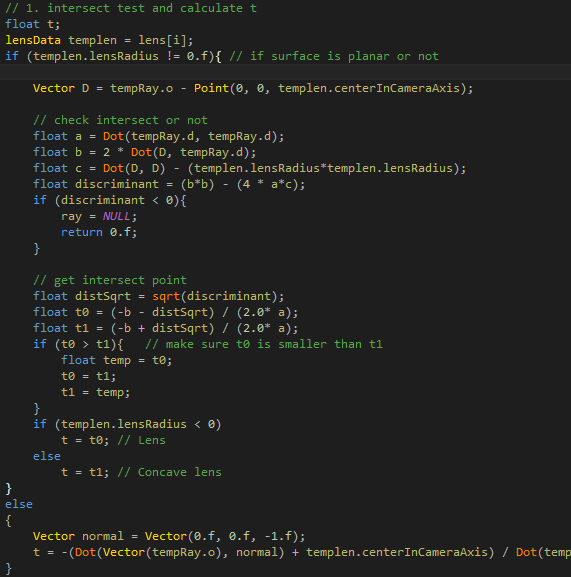
1. Ray tracing – GenerateRay()
   1. Generate a ray

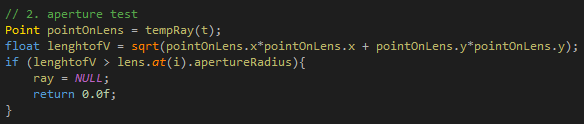
Create a ray by sample a point in the near lens and the point in the film in camera space

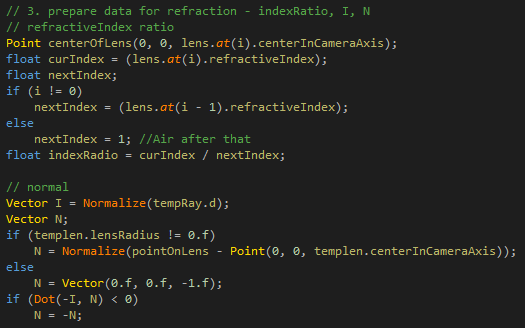


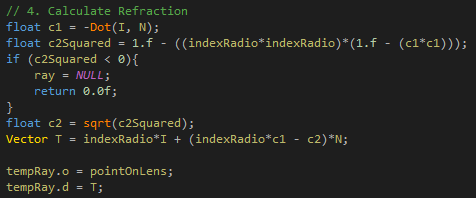
* 1. Ray-lens interaction

For each lens face element, from near to far, calculate the intersection between ray and lens, and change the ray origin and direction.









* 1. Return ray and weight

