

# CSC 473

## Program 5 – global illumination

**Due Thursday, June 2, at 11:55pm**

Overview:

This is the culminating assignment for the quarter for your rendering software. This assignment allows you to show off all the rendering capabilities and render a realistic scene using Monte Carlo Ray tracing. All your hard work in designing a complex piece of software using C++ will hopefully pay off in this assignment.

### Goals for assignment 5:

For this portion of your ray tracer, your program needs to:

- Support all prior rendering requirements (*but turn off anti-aliasing for GI renders*)
- compute global illumination using Monte Carlo ray tracing
  - please use **128** cosine weighted sample rays on your hemisphere for the first bounce
  - use **64-32** cosine weighted sample rays on your hemisphere for the second bounce
- Extra Credit: A feature of your choice from: texture mapping, motion blur, depth of field (or an instructor approved alternative – ask me)

For our final rendered scene, use the simple-gi.pov file (and optionally you can use the modified ‘Cornel Box’ on the website). You will again need to create your own ‘pretty’ scene that you render and submit the .pov file for.

---

What you should hand in via polylearn:

- Your code, include all files necessary to compile and run your ray tracer, including a Makefile or cmakeLists.txt
- A README.txt file with any information about what is working or not working with your implementation to assist the grader in determining what is causing potential errors in your output and help in assigning partial credit. Along with an expected runtime to compute a render of simple-gi.pov at 640 480 with no anti-aliasing.
- ***Your own .pov file and rendered image of an interesting scene***

You need to handin your code and images generated using poly learn. Look for the assignment directory.

### Grading breakdown (will be applied to each part appropriately):

30 all prior requirements working

55 Monte-Carlo ray tracing as evidenced by color bleeding

15 general sanity

Extra Credit: + 20 added feature of your choice, texture mapping, motion blur, depth of field (or an instructor approved alternative – ask me)

*Use the same command line assumptions from program 4 except add a final flag to turn on and off global illumination (1 on 0 off). For example:*

Thus:

**raytrace 640 480 sample.pov 0 1**

will render a 640x480 image file, “sample.tga” consisting of the scene defined in “sample.pov” using the Blinn-Phong *and NO anti-aliasing, but global illumination!* Unless you strongly want to change your default BRDF, please explain if this is true in your readme.