

Assignment #4

Point Value: 100 points

Due Date: March 21, 11pm

CS 184: Foundations of Computer Graphics

Spring 2008

Prof. James O'Brien

page 1 of 3

For this assignment you will write a ray tracer and use it to generate beautiful pictures that will astound all your friends.

1. This assignment is due as indicted above. Projects turned in late will lose points as described in the policies handout. This assignment should be done alone or in pairs. You may share ideas with other groups, but you may not share code.
2. You may develop on Unix, OS X, or Windows. The platform you use will be the one used to grade assignments. Keep in mind that there are slight variations due to OS versions, different libraries, and other factors, so you should verify that your code runs on the instructional machines appropriate for you platform choice.
3. We will be using the submit software for submission of this assignment. Instructions for using the submission software are [here](#). You should include a README file that at the minimum contains the following information:
 - Your (and your partner's) name
 - The platform your code runs on
 - The location of your source code (i.e. indicate who in your group has done the submission, and on what platform). Only one of the people in your group should submit the actual code. The other people should only submit the README file.

All files needed to compile your code should appear in the submitted directory. It is your responsibility to make sure that they will compile and run properly.

- Windows: The grader should be able to recompile your program by simply opening the project and rebuilding it from scratch.
- Unix and OS X: The grader should be able to recompile your program simply by typing "make".

You will also turn in some images. These should be named "image-nn.xxx" where nn is a number (e.g. 01, 02, 03...) and xxx is an appropriate extension (e.g. tif, jpg, ppm, etc.) The main input files needed to render those images should be named "input-nn". Finally there should be a "notes-nn" file for each input stating: the command line used to produce image-nn, how long it took to run, what features are demonstrated by the image, any other comments you'd like to add about the image.

4. Once you have your assignment working, you should also update your class web page to include an "Assignment 4" link to a page with some images generated by your code. Make sure that your images demonstrate all the features that you have implemented in your code.
 5. This assignment will be graded by looking at your web page to see that you have posted examples demonstrating all required features. The images on your website should correspond to the ones you submitted. You may add extra images to your website after the deadline, but
-

Assignment #4

Point Value: 100 points

Due Date: March 21, 11pm

CS 184: Foundations of Computer Graphics

Spring 2008

Prof. James O'Brien

page 2 of 3

these images should clearly be marked as extra images that were not included in the submission.

If you work in a group, you should all link to the same web page and the web page should list your group members.

Grading will include points for aesthetics and creativity as demonstrated on your web page.

6. ***Do not wait until the last minute to start this assignment.*** Even a minor bug in a ray tracer typically produces a black image with no other clue about what is wrong... that makes them very hard to debug. If you don't give yourself enough time, you will be most unhappy.

Check the news group regularly for updates on the assignment or other clarification. We will assume that anything posted there is henceforth known to all.

7. Submitting an image that was not generated by your code is considered cheating. Because raytracers may take a long time to generate a given image we can only spot check and you are largely on your honor that the images you show are yours. Please don't violate this trust.

8. For this assignment, you will write a ray tracer that can at least:

- Render ellipsoids (axis aligned ellipsoids are sufficient, but you must be able to apply transformations).
- Render polygons (triangles are sufficient)
- Use simple Phong Shading (in color)
- Compute shadows
- Compute reflections
- Use point and directional lights
- Use some reasonable method for accelerating ray tests (e.g. BSP trees or AABs)
- Write its output to a standard image format such as jpg, ppm, png, or tif.

9. Optional features that you can implement for extra credit are:

- Transparency with refraction
 - Anti-aliasing
 - Lens effects / depth of field
 - Super quadrics
-

Assignment #4

Point Value: 100 points

Due Date: March 21, 11pm

CS 184: Foundations of Computer Graphics

Spring 2008

Prof. James O'Brien

page 3 of 3

-
- Programmable shading
 - Texture, bump, and/or displacement mapping
 - Spot lights and/or area lights
 - Other *interesting* features

All features should be clearly documented in your README file and demonstrated in your example images.

10. You will use your program to render 5 or more example scenes that demonstrate the features you have implemented. You will turn in the images rendered by your code, the code, and some documentation (see above). Your images must demonstrate the features you implemented or you cannot get points for those features. You will also receive points for the aesthetics and creativity demonstrated by your images.

To demonstrate the use of accelerated ray tests you must have at least one image that involved 1000 or more inter-reflecting primitives.

11. The input file can be in any format you like. You are responsible for coming up with something that works and creating inputs for interesting images. An input may consist of multiple files.

Your output needs to be in a standard image format that most image viewers can read. Suggested formats are: JPEG, TIFF, PNG or PPM. Do not use an indexed color format like GIF. You can write your own code to write the files or you can use a standard library. PPM is very easy to write, but offers no compression. JPEG, TIFF, and PNG are supported by many standard libraries. (libTIFF, libPNG, ImageMagik, and many others) You are responsible for figuring out how to use these libraries.

Your images should be at least 640x480 and no more than 2000x2000.

12. This assignment is purposefully open ended. I am continually impressed by what Berkeley undergrads can do when given a bit of freedom, so here it is. You are encouraged to be creative and enjoy this assignment.

13. Questions should be posted to the news group or emailed to cs184.
