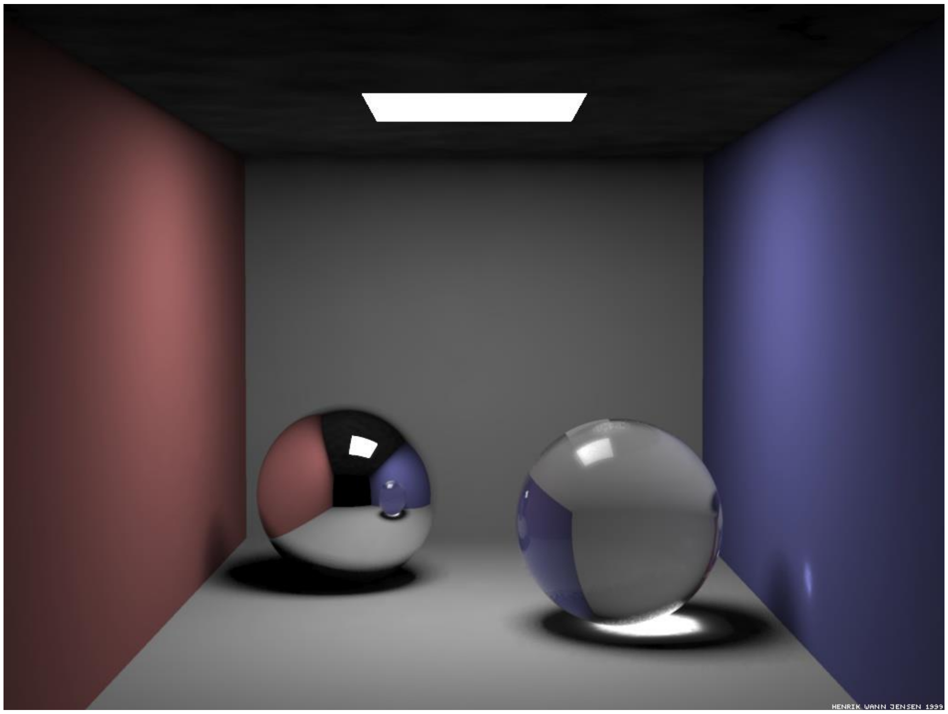
CSC 305 Assignment 1 – Ray Tracer

Due Sunday January 29th, 11:55pm



An example of a high-quality ray-traced image. Image credit: Dr. Henrik Wann Jensen, UCSD. 1999.

Overview

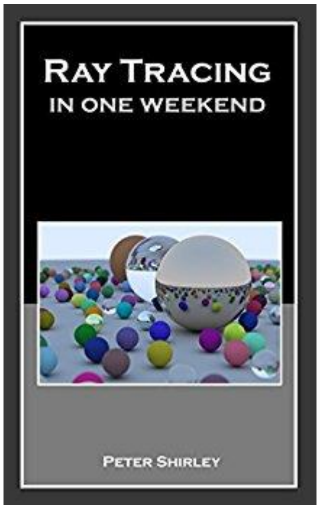
For this assignment, you will write a ray tracer using C++. The goal of this assignment is to familiarize yourself with the C++ programming language, and to learn the fundamentals of ray tracing and rendering.

The assignment follows the curriculum of Peter Shirley's "Ray Tracing in One Weekend" book series. We expect you to buy this e-textbook (and optionally its sequel, "Ray Tracing: The Next Week"). The books are $3 each, so this is hopefully not a problem. If you can't use Amazon for whatever reason, tell a TA about it and we'll see what we can do. You don't have to follow this specific curriculum, but please cite your sources if you decide to use another one.

You may base your code on the code samples in the books. You should not blindly copy code from the book, and you will not receive points if you use code without being able to explain it. Think about it as a study of the book, with an experimental part (your code) and a written part (your report). The report should be only a few pages, not a thesis.

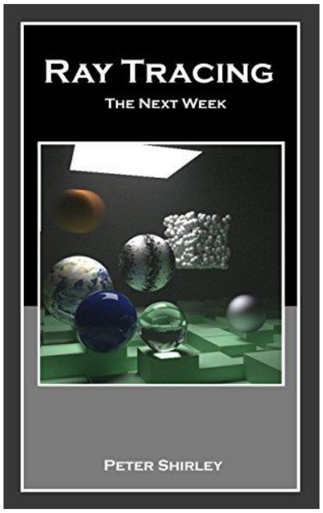
Standard Requirements (70%)

- Output an image
- Make your own vec3 class
- Render a background
- Render a sphere
- Render a plane (eg. floor under the sphere)
    - Do **not** do this by rendering a big sphere as the floor.
- Render multiple spheres
- Implement anti-aliasing
- Diffuse material
- Positional camera



Advanced Requirements (30%)

- Metal material
- Dielectric material
- Defocus blur
- Rectangles and Rectangular Lights
- Instances



Bonus Features (up to 10% bonus)

- Motion blur
- Bounding Volume Hierarchy (BVH)
- Solid Textures
- Perlin Noise
- Image Texture Mapping
- Volumes
- Triangle mesh

Standard requirements (and some advanced) are covered in the “Ray Tracing in One Weekend” book:

https://www.amazon.ca/Ray-Tracing-Weekend-Minibooks-Book-ebook/dp/B01B5AODD8

Advanced requirements and all bonus features are covered in the “Ray Tracing: The Next Week” book:

https://www.amazon.ca/Ray-Tracing-Next-Week-Minibooks-ebook/dp/B01CO7PQ8C

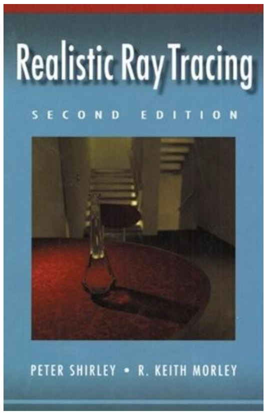
Please also provide a report explaining the features implemented in your program in your own words with your submission. You will not be credited for features which you cannot explain. Please explain the “what” “why” and “how”.

Please also provide screenshots that demonstrate the features you implemented. Explain the contents of each screenshot in your report.

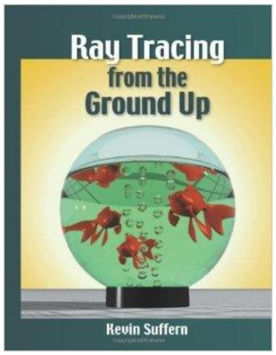
Additional Resources

You can find explanations for how to implement the features of this assignment in many books and on many web sites. This section contains some recommended additional resources, on top of the course textbook and lecture notes.

"**Realistic Ray Tracing**" by Peter Shirley is a classic. It's a bit old-school at this point, but still a good reference. It has chapters on ray-object intersections, sampling/filtering, viewing, triangle meshes, instancing with transformation matrices, and lighting methods. It includes example C++ code too, though it is quite old-school C++. There's a copy of this book in the UVic library, and we also have a copy in the graphics lab (ECS 330).

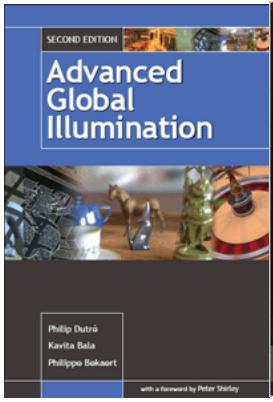


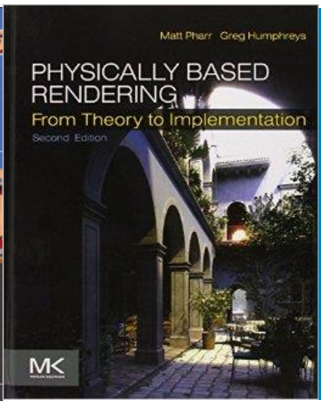
"**Ray tracing from the ground up**" by Kevin Suffern is also a great book to learn how to build a ray tracer. It's thorough, illustrated well, and also has example C++ code. You can find it in the UVic library.

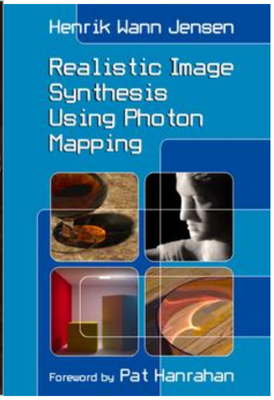


For more information on ray tracing, consider the following texts:

- “Advanced Global Illumination” by Philip Dutré, Kavita Bala, and Philippe Bekaert
- “Physically Based Rendering, From Theory to Implementation” by Matt Pharr and Greg Humphreys
- “Realistic Image Synthesis Using Photon Mapping” by Henrik Wann Jensen


SECOND EDITION
Advanced Global Illumination
Philip Dutré
Kavita Bala
Philippe Bekaert
with a foreword by Peter Shirley


Matt Pharr Greg Humphreys
PHYSICALLY BASED RENDERING
From Theory to Implementation
Second Edition
MK



Useful Third Party Libraries

You may find the following third party libraries useful. You don't have to use them.

- GLM: A C++ library for common vector/matrix operations.
  Please don't use its built-in ray intersection functions, you have to write those yourself.
  http://glm.g-truc.net/
- "stb_image_write": An easy-to-use C library for writing PNG/BMP/TGA/HDR images.
  http://nothings.org/stb/stb_image_write.h
- "stb_image": An easy-to-use C library for reading images.
  http://nothings.org/stb_image.h

Submission

Please submit your assignment through Connex. In your submission, upload a zip file that contains the code for your assignment project, with your report and screenshots as explained earlier in this document.

Do not submit any compilation artifacts, such as ".o" files, ".pdb" files, or ".exe" files.