COMP 356: Computer Graphics
Final Project

For your final project, add the following functionality to the ray-tracer you developed in Homework 2B:

(1) A bounding-box tree surface type.
(2) Support for transparent objects with refraction, as described in Chapter 13.1 of Shirley and Marschner.

A few comments:

(1) I have provided a few files on the website that you must use: `surfaces_lights.h`, which specifies functions for defining the scene and lighting; `surface.h`, which specifies the functions and types relating to surfaces; and `surface.c`, which is a partial implementation of `surface.h`. You may not modify the `.h` files, and you must include these three files as part of your submission.
(2) Your implementation of `surfaces_lights.h` must be named `surfaces_lights.c`, but you must *not* include this file in your submission (it will be overwritten by my version).
(3) I strongly recommend implementing the bounding-box tree surface type before working on transparency. As a point of comparison, a test scene involving a transparent sphere on a chess board took about 2.2 seconds to render using a bounding-box tree, but 15.1 seconds without.
(4) The only "gotcha" not discussed in the text that I discovered in implementing transparency has to do with shadows cast by transparent surfaces. They should cast shadows, because otherwise they appear to float in mid-air. But casting an opaque shadow just looks wrong. You will have to come up with an appropriate compromise.
(5) Make sure your ray-surface intersection routines are all correct!
(6) I will not put any transparent surfaces into a bounding-box tree surface.
(7) Be certain to clearly document where you have added your code so that it is easy for me to find it.