Ray Tracing (Do you want to ray trace a snowman?)

This assignment will be 25 of the 55 points available for the programming assignments.

Notes for CSE 470
Instructor: Ross Maciejewski

1 Project Guidelines

- Have fun, learn, be creative, and create something you are proud to show-off!
- No teamwork allowed! It is ok to discuss the projects with other students. You can use the discussion board on myASU (The TA will also try to answer your questions posted at myASU, but only sporadically); however your code must be your own. (You would be surprised how easy it is to detect copied code!). You cannot post code on the discussion board! (only very small code sections)
- Comment your code - you will be glad you did. You will be reusing code throughout the semester.
- You will upload your projects on myASU. Details about this will be announced in class.
- Late projects will loose points:

  10% for up to 24 hours too late.
  20% for up to 4 * 24 hours too late.
  The myASU timestamp will show when you uploaded your project.
  40% for one week late
  75% for two weeks late
- If you have **good reasons** that you cannot complete a project on time **and** you have written documentation, then we can make adjustments to due dates. However, you must notify us before the due date that you would like to discuss such an arrangement. Good reasons would be illness, family emergency, visiting a conference to present a paper, …
- All the University, Student Life and Fulton School academic integrity policies hold: *http://www.asu.edu/studentlife/judicial/integrity.html*

2 Upload your projects

How to upload a Project
This is a short guide for the successful upload of a project:

- Check the due date of the project on myASU
- Use myASU (digital dropbox) to upload your project. (Do not send the program as email attachment to the instructor or TA!)
- Your project should be packed in a zip file and named with the following naming convention:
  PX.Lastname.V1.zip
  (if you submit some updated files, use a different version)
  So my second project would be named
  P2.Maciejewski.V1.zip
  If I then found a last minute mistake and want to upload again I would use the name P2.Maciejewski.V2.zip
- If there is someone with the same last name in the course you can add your first name after the last name in the filename.
- The zip file should include all the source code and should be ready to compile. That means it should include your Visual C++ project files.
- The zip file should include an executable and the freeglut.dll. You have to make sure that the executable can be started directly by clicking on it. (from wherever the file is downloaded to). You might have to copy the exe file from the Release folder into the project directory.
- A short text file called “instructions.txt” – this text file should include the following:
  - a) instructions about how to navigate in your program, including the function of special keys (e.g. x exits the program)
  - b) special functions that you implemented
  - c) changes from the original specification that you negotiated with the instructor or TA to the original specification
  - d) other things you consider important
  - e) specify the libraries or code that you used and did not write yourself. You do not need to specify code that we provide on myASU.

3 Project 4

3.1 Purpose

This project has been designed for you to get experience with ray tracing.

3.2 Overview

In this assignment, you will be building a ray tracer. As already discussed in lecture, ray tracing is a powerful method to perform global illumination. Ray tracers are widely used to render photorealistic images and animations. You can try the interactive ray tracer (http://www.cs.cmu.edu/~efros/java/tracer/tracer.html) to see how the ray tracer works step by step.

At the end of this assignment, your ray tracer should be able to minimally handle opaque surfaces with lighting, shadows, reflections and texture mapping. Provided for you will be some starter code that performs some elementary parsing functions.

Here are the elements that your program must have. Despite this "laundry list" of requirements, there is some room for creativity! Note: If the text suggests a menu item "Sound - Off" that would mean that "Off" is an entry in the submenu "Sound".

- **10 pts –** You should use a large polygon for the walls, floor and ceiling, essentially you are in a closed room. You should give the floor a checkerboard tile pattern using a procedural texture or an image. Use an appropriate number of squares so that it looks like tiles. You should give the walls striped wall paper.

- **5 pts –** In the middle of the room you should create a snowman comprised of three spheres of different colors.

- **5 pts –** In the center of two walls you will place two triangle mirrors that will be partially metallic (brushed brass).

- **10 pts –**You should read the eye-point and the lookat point from the xml file (test.xml) as provided in the sample code and render your scene using a perspective camera.

- **10 pts –** You should allow the user to move the camera as in Assignment 3. The camera is controlled as follows: You can go forward/backward with the cursor up/down keys or rotate around the y-axis using the left and right keys. (You have to check that you are not outside of the room too).

- **40 pts** – You should include Phong lighting with shadows. If I press 1 on the keyboard, there will be one light directly above the snowman. If I press 2 there will be two lights each at the lower front corner of the room (a room has 4 corners on the floor, pick two that are next to each other for the light position).

- **20 pts – Recursive ray tracing: (Graduate Students Only)**
  The mirrors placed in the scene are now metallic and reflective. You must set up the scene to correctly handle reflection.

- **Extra Credit:**
  You may earn up to 10% extra credit on your semester programming grade by implementing the following:
  - Recursive Refraction: allow transparent surfaces that refract rays. All an ice ball in front of the snowman that lights will refract through.
  - Distribution Ray Tracing - Soft Shadows: cast more rays to create soft shadows

  Note, there will be no TA or Faculty support for the extra credit portion.

Grading

These points represent an initial weighting of the functionality. **Important:** You get full points for a *good* implementation of the required functionality. In principle you will get full points if you implement the requirements correctly, but there are many ways to do things according to specification that we might consider *unreasonable* or *undesired*. Especially we will deduct points if the design choices make it difficult to judge the program (E.g. you choose materials that are so dark that it hard to see anything, the objects move too slowly or too fast, the control of the program is extremely difficult.)

Note: we might change the weights slightly if a requirement is misunderstood by a majority of the class or some other special circumstance. I do not expect that to happen though.