Basic Ray Tracer

How to:

Go into libst and type command "make," type command "make" again in the directory that contains all my cpp files, and then type command "./raytracer <path from current directory to datafile>

Extensions:

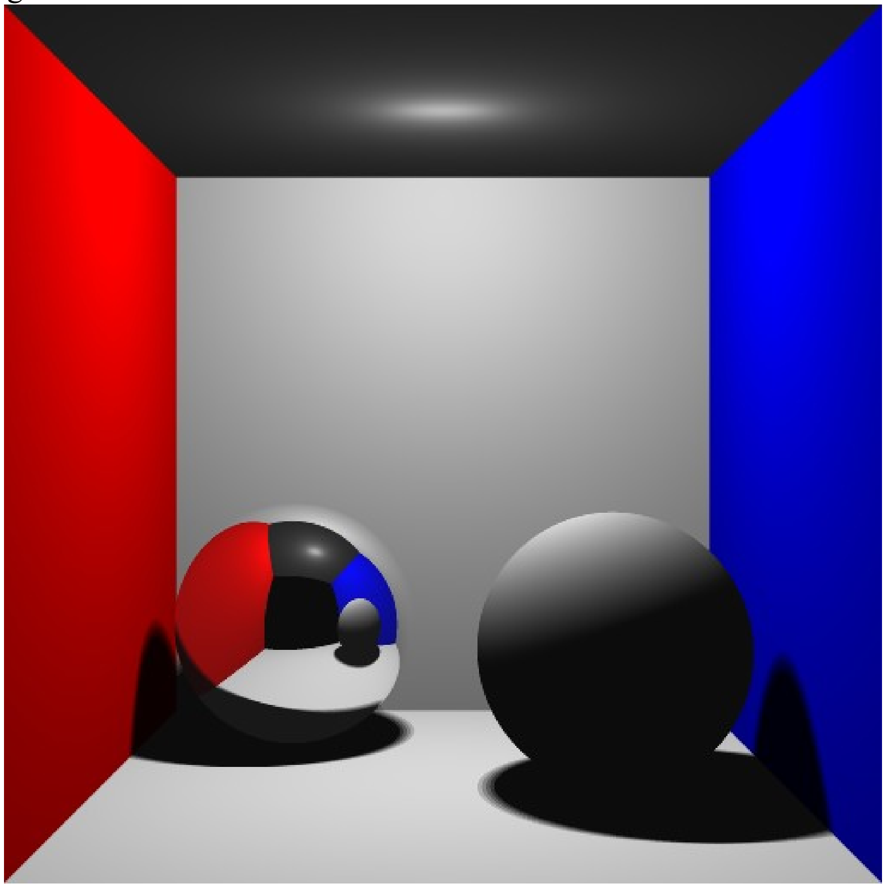
(1) Anti-aliasing

Raytracer anti-aliases automatically because I wrote it so. No way to turn off anti-aliasing!

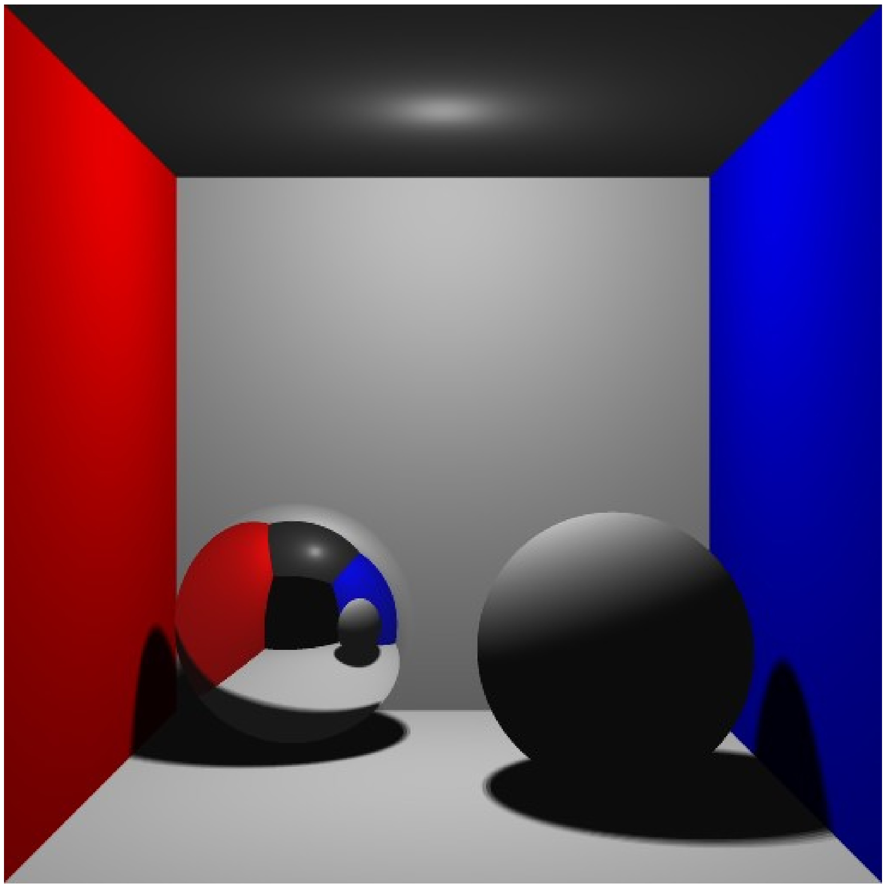
(2) Soft shadows using area lighting - line light, circle light

In data file, in a new line, type "LineLight <start point> <end point> <total intensity> <number of samples to pick from line light>" or "CircleLight <center point> <radius vector1> <radius vector2> <total intensity> <number of samples in the centermost line>" Warning: LineLights do not take much time, butCircleLights will take a very long time for each <# samples in the centermost line> that you increase.

Using one LineLight:

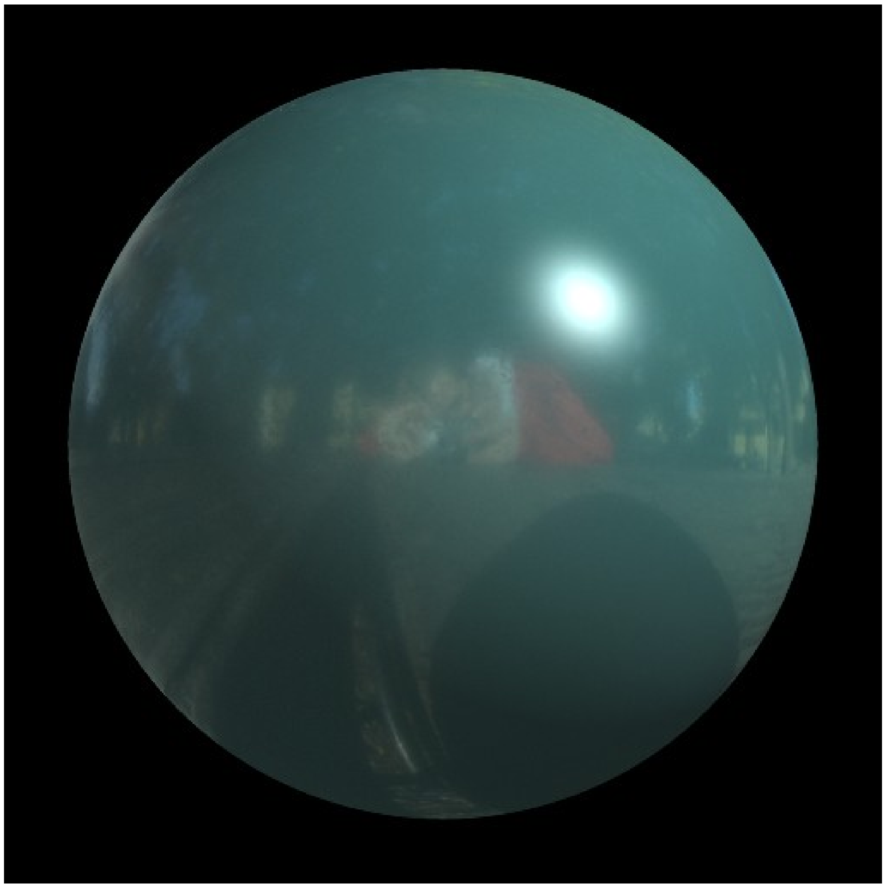


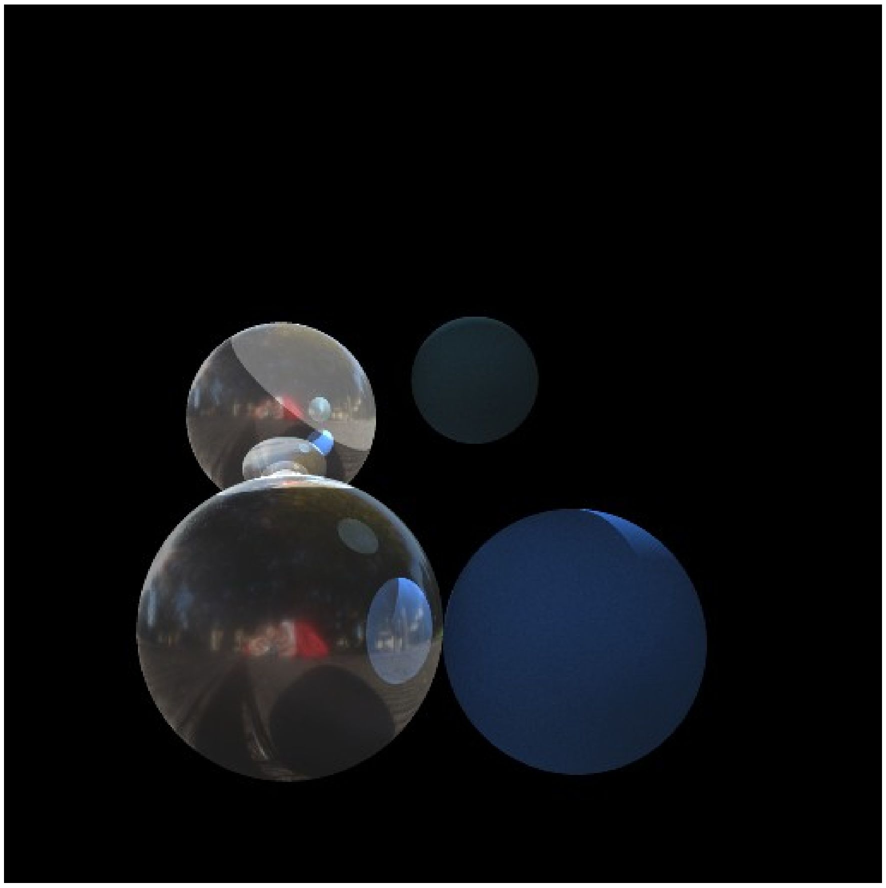
Using one CircleLight:



(3) Environment-mapped lighting

In data file, in a new line, type "EnvironmentLight <light probe image file> <intensity of lighting>". There can only be one environment light per scene. If this line is typed again below with different arguments, the old environment light will be overlooked. The environment lighting works in different ways for different reflectances. For mirror reflectance, it's just the reflectance*<whatever color that corresponds to the normal>. For diffuse and specular however, it samples a set number of faux-normals starting from the surface point, and applies diffuse and specular lighting calculations. Then it averages these and adds it to the overall color of the surface point being computed. I didn't have time to implement shadow-checking for each faux normal, but the following things work:



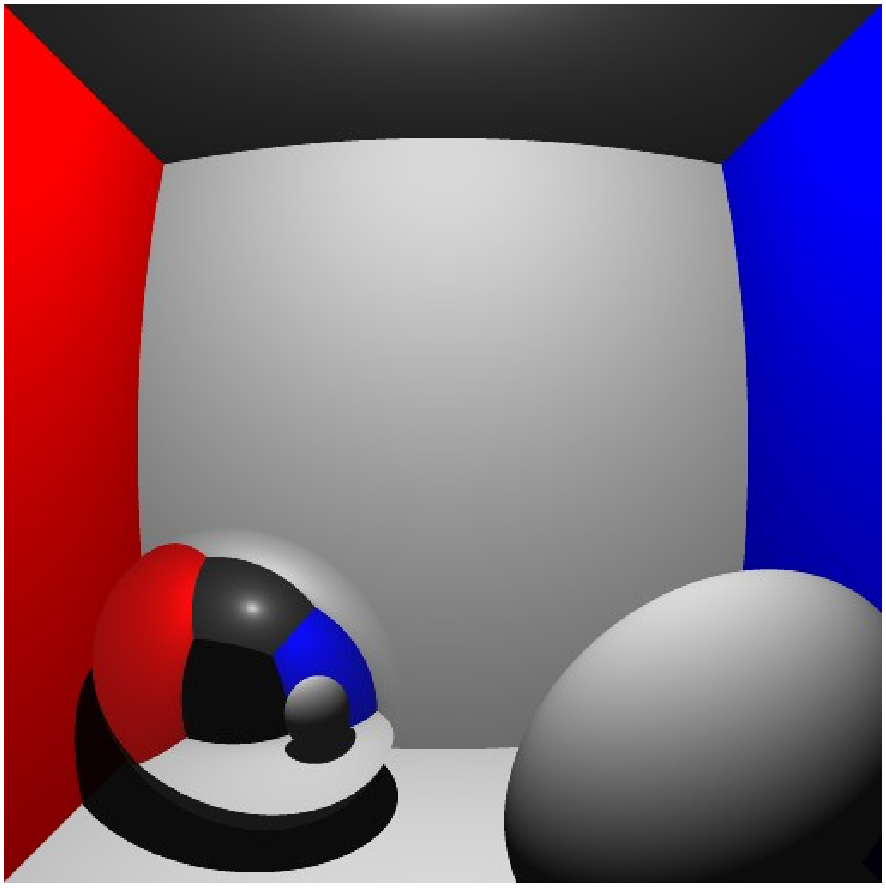


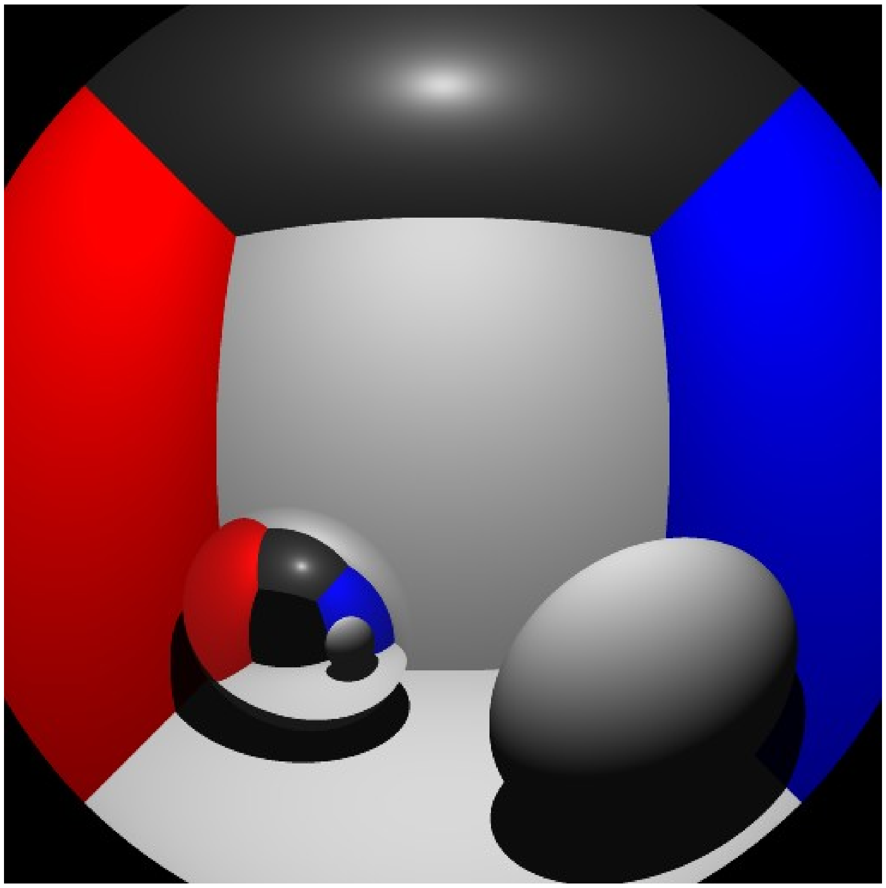
Notice how on the diffuse spheres, the surfaces looks more bumpy/noisy due to environment lighting compared to the previous images that were rendered with point lights.

(4) Fisheye lens, two types of cylinder lenses

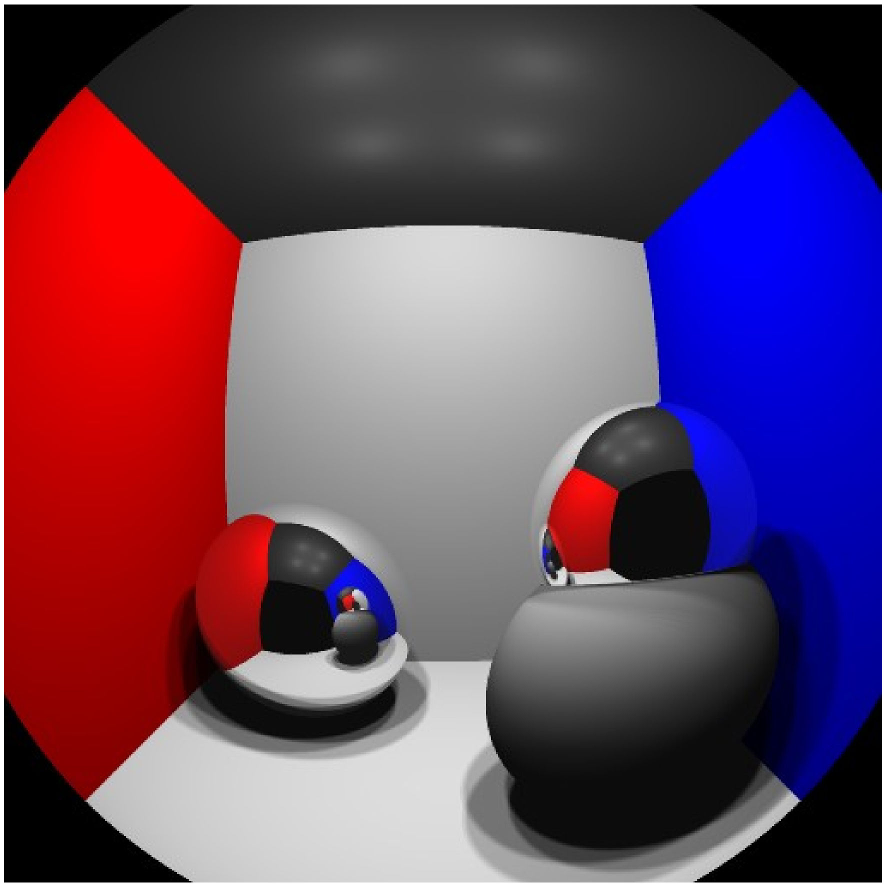
Type "Fisheye" in a new line in the data file to make the camera look through a fisheye lens. Type "Cylinderx" to make the lens a cylinder that curves in the x direction and "Cylindery" to make the lens a cylinder in the y direction. All the special lenses are curved away from the camera.

Using Fisheye:

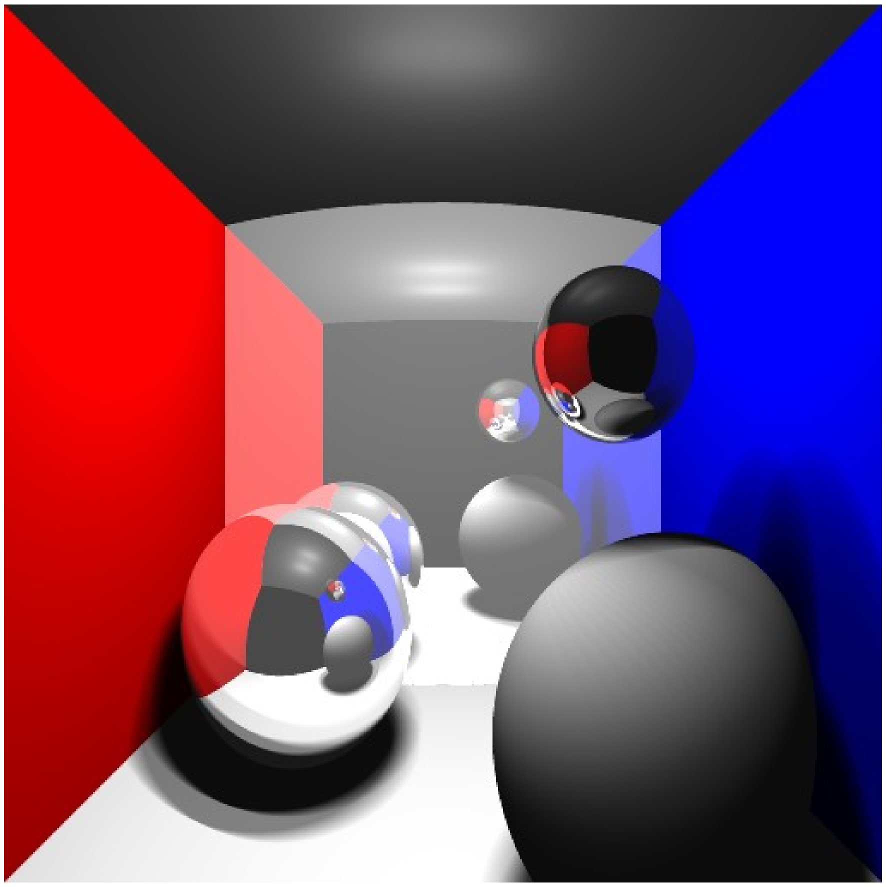




Using Fisheye and four CircleLights:

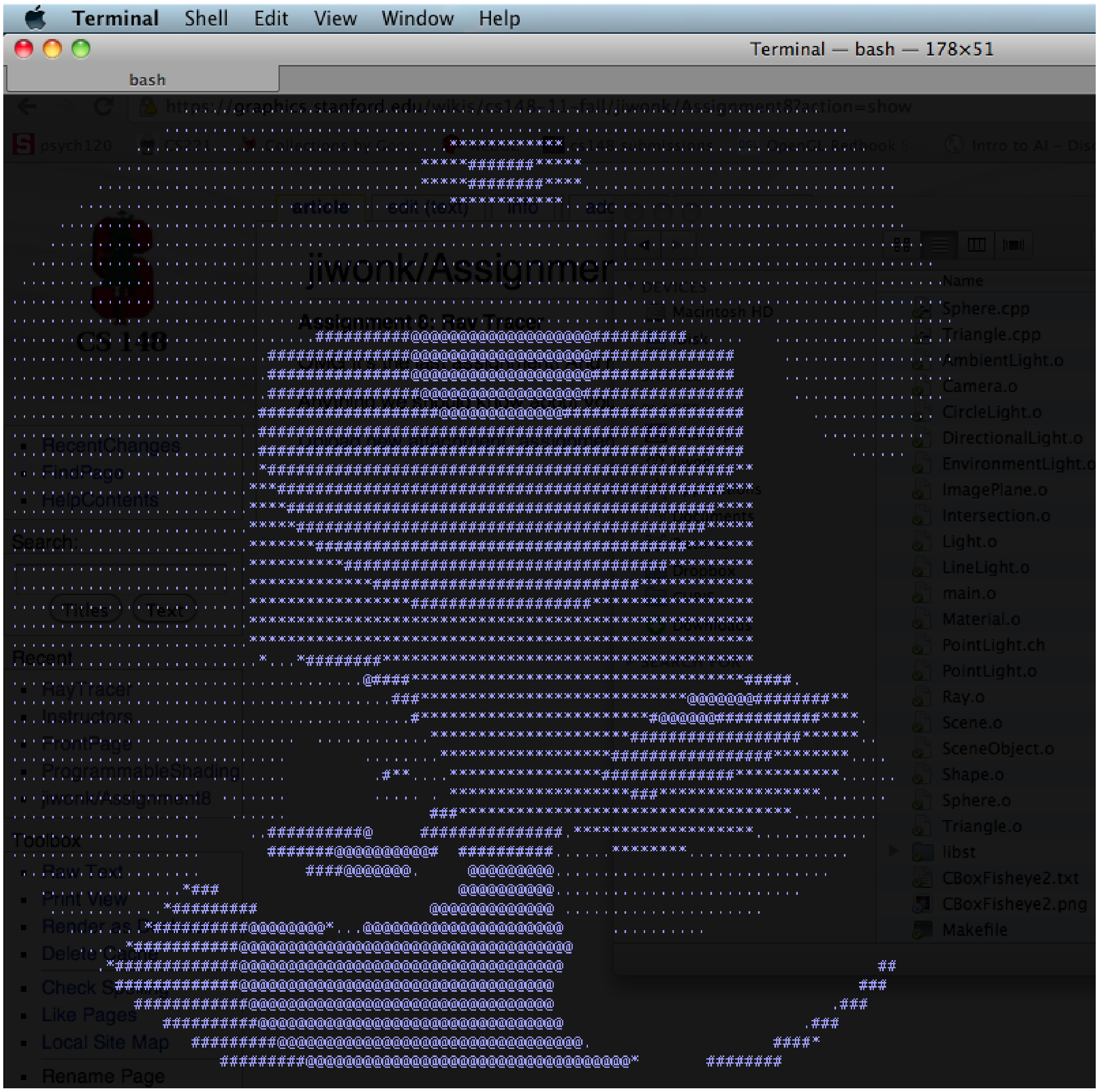


Using Cylinderx and two LineLights with the back wall as an almost-mirror surface:



(5) ASCII rendering

Since waiting for the image to render is a tedious job, renders the image line by line (skips some lines to make image fit on screen) in real-time, in text. Used the luminance equation from the image compression assignment.



Sphere.cpp
Triangle.cpp
AmbientLight.o
Camera.o
CircleLight.o
DirectionalLight.o
EnvironmentLight.o
ImagePlane.o
Intersection.o
Light.o
LineLight.o
main.o
Material.o
PointLight.ch
PointLight.o
Ray.o
Scene.o
SceneObject.o
Shape.o
Sphere.o
Triangle.o
libst
CBoxFisheye2.txt
CBoxFisheye2.png
Makefile