

Project 1



Project description

Attached Files: [proj1.pdf](#) (65.607 KB)



Center of a pixel

Attached Files: [cop.pdf](#) (45.729 KB)



Skeleton code

Attached Files: [Szymczak.tgz](#) (4.341 KB) Contains:

- a skeleton of a function that

reads the input file

- a simple class for writing pixels and saving them as a PPM file
- a README.txt file with a sample work plan

The format is exactly the same as what we expect to get from you as the final submission, except that you don't need to include the input.txt file.



Input file examples (this is NOT the test data that will be used for grading - don't include in your submission)

Attached Files: [inputs.tar](#) (960 KB) Unpack and click on the 'index.html' in the inputs directory.



Orientation of the image

Attached Files: [corners.tar](#) (10 KB)

You can use the enclosed

test input files to make sure your output is correctly oriented:

ll = sphere should appear in lower left corner

ul = ...upper left...

lr = ...lower right....

ur = ...upper right....

If you want to flip your image about the horizontal axis, change

`RGB &pix = img.pixel(x,y);`

to

`RGB &pix = img.pixel(x,resolution_y-1-y);`

Similarly, to flip about the vertical axis, replace this line with


`RGB &pix = img.pixel(resolution_x-1-x,y);`

To rotate by 180 degrees you need to superpose the two, i.e. replace the line with

`RGB &pix = img.pixel(resolution_x-1-x,resolution_y-1-y);`



Test data

Attached Files:  [testdata.tar](#) (220 KB) out*-1.png are the output images you should obtain for report #1

out*-2.png are suitable for report #2

out*-final.png are obtained with the complete ray tracer



Compiling the skeleton code under windows

Here is what worked for me (VC++6):

- (1) create a 'Hello world' console C++ project
- (2) replace the main() function with the contents of the raytrace.c file; don't remove the #include "stdafx.h" line!
- (3) create a directory and make it the working directory (Project->Settings)
- (4) Place an input file in the working directory and name it 'input.txt'

The code should now compile and run. It will save a PPM image 'output.ppm'. Look for it in the working directory.



Viewing PPM files under windows

There are plenty of free image viewers that support

the PPM format. Examples:

- IrfanView <http://www.irfanview.com>
- xnView <http://www.xnview.com/en/index.html>
- gimp <http://www.gimp.org/windows>



Huge input files

Attached Files:  [bigones.tgz](#) (13.533 MB) Don't take them seriously, some may take very long to run. The

images show results with shadows turned off.