COMP 5812M: Foundations of Modeling & Rendering 2019-2020

ASSIGNMENT 4: Blinn-Phong Raytracing [10 marks]

In this assignment, you will build a simple 3-D raytracer with exactly the same input as the rasteriser in Assignment 3. Your output should be nearly identical to that Assignment: it will not be identical, because we ignored perspective correction in the barycentric interpolation.

Your code should output a set of PPM files representing various stages of the ray-tracing process. For details of the PPM format, please see Appendix I or http://paulbourke.net/dataformats/ppm/. You should store a 128x128 image in PPM in 8 bit (1 byte) per channel RGB.

Your input data may be hard coded or read in from file. Since rendering multiple triangles is just a loop, this assignment will focus on rendering a single triangle, with the following attributes:

Vertex Position RGB UV

A (61, 10, 1) (255,0,0) (0.160268, 0.290086)

B (100, 100, 1) (0, 255, 0) (0.083611, 0.159907)

C (25, 90, 1) (0, 0, 255) (0.230169, 0.222781)

You should set the eye position at (0,0,0) looking in the direction (0,0,1) with up vector (0,1,0). Assume that your camera has a field of view of 90° (i.e. 45° each side of centre).

a) Set the image to light yellow (255, 255, 192), and render an image where any pixel whose ray intersects the triangle is set to black: this will test your geometric intersections. You can cross-check this by drawing the correct triangle by hand on a piece of graph paper. [2 marks]

b) Compute the barycentric coordinates of your point of intersection and use that to render the triangle in colour with the colours shown above. Note that perspective should cause your triangle to be coloured slightly differently from the one in Assignment 3. [2 marks]

c) Create four images for Blinn-Phong lighting: one each for ambient, diffuse and specular, and a final one with all three (we will ignore emissive lighting). Remember to use Phong (per-pixel) lighting, but that the triangle has a single normal vector to make life easier.

Choose reasonable material properties, albedos and coefficients and document your choices in the readme.txt. [4 marks]

d) Add a ground plane consisting of two triangles at x = 0 and render with shadow rays

[2 marks]

**PENALTIES:**

Poorly structured or badly commented code may be penalised by up to 25% of the marks available.

Poorly formatted output files may also be penalised by up to a further 25% of the marks available.

Code without a readme or makefile may be penalised by up to 10% of the marks available.

Code that does not compile properly will be assigned a mark of 0, but I will usually give the student one chance to correct this.

DUE DATE: Monday, December 2, 10:00 am