Computer Graphics Exercises

These exercises are grouped into three parts. Part I contains routine exercises to help you understand the ideas directly presented in lectures. The exercises in Part II are designed to grow and deepen your understanding of principles that underpin computer graphics. These exercises extend the lecture material and invite you to think about 'why' questions and about optimisations to what was lectured as well as alternatives to what was lectured. Part III are open-ended, stretching questions that you could tackle in your mini research project.

I hope you enjoy working through these questions, puzzles and research questions!

John Fawcett, July 2023

Part I

1. Use your favourite programming language to produce a P3 image file for a scene with a single cube:
   The front should be white: RGB=255,255,255
   The back should be yellow
   The left should be red
   The right should be blue
   The top should be green
   The bottom should be purple
   The centre should be at (40,30,20). Hint: scale up from a 1x1x1 cube to 39x39x39.
2. Modify your program to cast 16 rays through each pixel. Space the samples evenly so the rays should go through each pixel at 12.5%, 37.5%, 62.5%, 87.5% of the width and height of the pixels. Set the pixel to the average colour of the 16 samples.
3. Modify your program to cast about 100 rays through each pixel to render a depth of field. Each ray should come from position (randomly or deterministically) chosen to be near the camera, up to distance A away: Hint: try A = 0.05. Set the pixel to the average of its 100 rays.
4. Use Phong's equation to implement specular reflection in your ray tracer. Try to avoid the common mistakes:
   V is back towards the viewer from the point where the initial ray hit the cube.
   R is the reflection in the surface normal of a vector from the intersection point to the light.
   Hint: set a single light source at -80, -60, -30.
5. Implement spheres in your ray tracer
6. Implement cones and render an image of an ice-cream: a brown cone with a white hemi-sphere on top. Think about how to implement a hemi-sphere!
7. How would you implement partially transparent objects?
8. How would you add support for refraction (e.g. air/water interfaces, air/glass interfaces)?

Part II

1. Try to implement a torus primitive object (a doughnut shape). This is just maths but it's a bit more difficult than cubes, spheres and cones.
2. Implement coloured lights in your ray tracer (e.g. a blue object under red light will appear black).

3. We have used RGB colours throughout this lecture. Why do you think we did that? What other colour gamuts might we have used – and why?

Part III

1. Implement area light sources (not point sources of light). This will give you beautiful soft shadows in your images.
2. Read up on image maps, bump mapping, and displacement mapping. How would you implement these features?
3. Does it make sense to shine “black light” into an image? What should the result be?