1 Before attempting the problems

This exercise sheet concerns the latter half of the course: the details of the rendering pipeline in OpenGL, textures, practical rendering issues, fundamentals of colour theory. Once again, this exercise sheet does not contain any coding assignments.

A nice thing about this course is that every problem you come across can be visualised and intuitively understood. The solutions themselves are varied and there are many more than are presented in the course (the history of development of these techniques is really interesting). The mathematical background and linear algebra require developing some intuition – hopefully this will be provided to you through Maths courses.

2 Problems

1. (a) What are the advantages of GPUs over CPUs and why are they better suited for graphics rendering?
   (b) What is the difference between OpenGL and GLSL?
   (c) Give a brief high-level overview of the OpenGL rendering pipeline and all of its steps.
   (d) Are the front and back clipping planes of a viewing frustum both necessary?
   (e) In what step is the z-buffer algorithm relevant?
2. (a) What is the difference between bump and displacement maps? Why would you use one over the other?
   (b) Explain the purpose of MIP maps.
(c) How can you use textures to render the sky and clouds?

3. Find the UV map for projecting a texture onto a unit sphere centred at the origin. Are there any mathematical and practical issues with doing this?

4. Decimation is a process of reducing the number of polygons of a model, while still remaining sufficiently similar to the original.\footnote{In Blender this is done by adding a Modifier to an Object.}

In 3d animated films and especially in video games, it is common to have several 3d models for the same entity with different levels of detail (LOD).\footnote{Versions 4.x of OpenGL add a Tessellation shader between the vertex and fragment shaders, which can be used to create different LODs – though you would start with a low-poly model and construct a high-poly one from it.} For characters these will usually be created using a more manual approach, whereas for environment art (especially terrain) it is more common to make heavy use of automated decimation tools. Typically you would have at least three or four LODs for terrain.

(a) Why might it be useful to load the same object at different LODs?

(b) What is the implication for textures?

(c) Describe a decimation algorithm that takes as input a 3d mesh formed out of \( n \) triangles and a number \( p \), where \( 0 < p < 1 \), and outputs a mesh made out of approximately \( np \) triangles.

5. Describe the problems of flicker and tearing. Give an overview and describe the techniques used to solve these issues.

6. You are implementing a graphics engine for a survival horror game. Your task is to add fog into some scenes. The model that you use is that at distance 0 from the viewer there is no change in visibility, at distance \( d \) the fog totally takes over, the colour of the fog is \( c \), and an exponential increase in fog strength holds in between 0 and \( d \): the fog density \( f \) (‘proportion of fog’) at distance \( x \) follows a law of the form:

\[
 f(x) = a + b \exp(\lambda x), \quad 0 < x < d
\]

for a set parameter \( \lambda \). Find the values of \( a \) and \( b \) and give a sketch of what needs to be done in OpenGL code (in setup and the shaders) to model this.

7. Many computer vision algorithms (edge and object detection, motion tracking, inferring depth, image recognition) tend to convert RGB images into HSV at the start and work with that representation. Why might this be the case?

8. (a) What is metamerism?

(b) What is the difference between luma and luminance?

(c) Why does sigmoidal tone mapping make sense?

(d) Compare and contrast glare correction and gamma correction.