Exam
EDAF80 Computer Graphics : Introduction to 3D

2020–04–14, 14.00–19.00

Please answer in English.
Please answer each question on a separate page.
Dictionaries for English are allowed.
Electronic calculators are **not** allowed.
This is a closed book exam, so no other resources are allowed to be used.
When you have completed the exam, or at the latest 19.00, email your answers, as pdf, or jpegs, pngs, etc to michael.doggett@cs.lth.se.
Please make sure you name is on every page/image you send.
You exam material must be received by 19.15.
Grading: The maximum score is 6.0. A score of 3.0 or above is needed to pass.

1. Transformations

   a) Given the three matrices **A**: translation by the vector $\mathbf{v} = (2, 8, 1)$, **B**: rotation 90 degrees around the x-axis and **C**: a non-uniform scaling with 6 in x, 4 in y and 7 in z. Given the point $P = (1, 1, 1, 1)$, what is the location of $P' = \mathbf{BAC}P$? *(0.3p)*

   b) Why can normals not necessarily be transformed with the same matrix as points? How should they be transformed? *(0.2p)*

   c) What is backface culling and how is it determined? *(0.2p)*

   d) Describe the transform **M** needed to move the triangle A to A′ and the transform **N** that moves B to B′ in Figure 1 below. You can use the syntax used for the Node class in Assignment 1, e.g. Translate(x, y), RotationZ(θ), Scale(x, y). Include in your answer the order of operations. Assume that the origin is placed at the lower left corner of triangle A, and all operations operate in the XY plane. *(0.3p)*

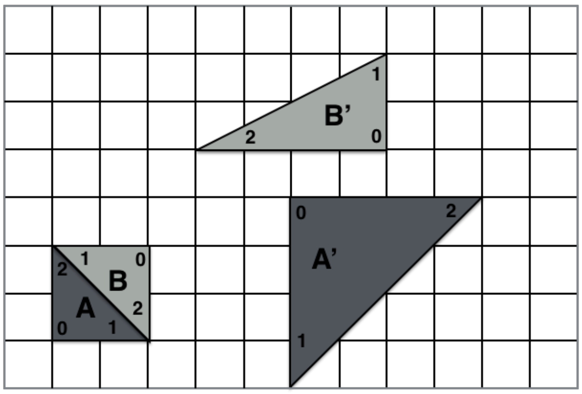

Figure 1: Transform from A to A′ and B to B′

2. Bump Mapping and Coordinate Systems

   a) Explain the bump mapping algorithm. *(0.4p)*

b) How would you compute the direction vector for an object that follows a cubic curve and points in the direction of the curve? *(0.3p)*

c) Describe three coordinate systems (spaces) used in the graphics pipeline. Include in your answer other possible names for the system, its use and possible construction. *(0.3p)*

3. Shading, GLSL

a) Using the view vector V, and the light vector L, explain how the specular component of Phong Shading is computed? Compute and include any other variables that are necessary. *(0.2p)*

b) What is a cube map and what can it be used for? *(0.2p)*

c) The OpenGL Shading Language (GLSL) includes the `texture(S,I)` command. Give two different examples of the command's usage, and explain what the two parameters are for each usage of the command. *(0.2p)*

d) Describe what the GLSL type qualifiers, `uniform` and `const` do, and how they are used in GLSL. What is the difference between them? *(0.2p)*

e) GLSL has several built-in functions. One is the smoothstep function. Give an example of how you could use this function for shading. *(0.1p)*

f) What variable must a Vertex Shader set, and what does it represent? *(0.1p)*

4. Rendering, Hierarchical Modeling, and Projections

a) Name two important differences between rendering 3D images for Real-Time applications like computer games, and for Offline applications like feature films. *(0.2p)*

b) What is the difference between spin and orbit? Explain the two concepts. *(0.2p)*

c) When do you use a $4x4$ matrix, as opposed to a 3x3 matrix, and why? *(0.3p)*

d) Explain the perspective transform. What is it? How does it work? How is it used and set up in OpenGL? *(0.3p)*

5. Graphics Pipeline

a) Give the three main stages of the Graphics Pipeline and describe what happens to a triangle as it passes through these stages. Include in your answer what inputs and outputs are generated. *(0.3p)*

b) Different scenes and shading techniques place varying workloads on the stages of the Graphics Pipeline. Give an example of a scene that places a heavy load on the Vertex Shader. Give another example of a scene that places a heavy load on the Pixel Shader. *(0.3p)*

c) Compare how rasterization and ray tracing determine visibility per pixel. Include in your answer the difference in how pixels and triangles are processed in the two algorithms. *(0.4p)*

6. General

a) The rendering equation is given as

$$I(\mathbf{p}_c, \mathbf{p}_s) = v(\mathbf{p}_c, \mathbf{p}_s)\left[\epsilon(\mathbf{p}_c, \mathbf{p}_s) + \int_S \rho(\mathbf{p}_c, \mathbf{p}_s, \mathbf{p}) I(\mathbf{p}_s, \mathbf{p}) d\mathbf{p}\right]$$

Why is the equation hard to solve? *(0.1p)*

b) Show how the rendering equation can be approximated using the phong shading model. Discuss some assumptions, and how terms are replaced to simplify the equation to phong shading. *(0.3p)*

c) How would you attach the camera to an object in a scene? For example, a camera that is attached to a planet in a solar system and moves around with the planet as it orbits the sun. Explain the steps that would need to be taken to achieve this within the bonobo framework used in the first assignment to make a solar system taking into account the effect of hierarchical modelling. Include the values that are needed, how they are computed, and how they are used to set up the camera for rendering. *(0.3p)*

d) What range of values is used to store normals in a Tangent Space Normal Map used for Bump Mapping? What range is this converted to before use in a Bump Map shader? What assumptions could be made about the possible range of the channels of the normal, to improve the precision of the stored normal value? Are there any other possible methods to improve the precision? *(0.3p)*

The end.