TRINITY COLLEGE DUBLIN

THE UNIVERSITY OF DUBLIN

Faculty of Engineering, Mathematics and Science

School of Computer Science and Statistics

SS Mathematics **Trinity Term 2015**

CS4052 – Computer Graphics

Tuesday 28th April, 2015 **Luce Upper** **09:30 – 11:30**

Dr Anton Gerdelan

Instructions to Candidates:

Answer any FOUR questions – 25 marks each.
All questions carry equal marks.

Please use a **separate answer book** for each question.
The entire question paper must be handed in at the end of the examination.

Question 1: Illumination

(a)	Draw a diagram of the two vectors used in the diffuse component of Phong lighting.	(5 Marks)
(b)	Briefly explain the difference between flat (per-facet) and Gouraud shading models.	(5 Marks)
(c)	Write a shader (you may use pseudo-code) that calculates ambient and diffuse reflection models for a point light. You may ignore specular reflection. Assume you have the following variables: • any matrices required; • surface normal in eye coordinates; • surface position in eye coordinates; • surface diffuse and ambient reflection constants; • light position in world coordinates; • light diffuse and ambient colours.	(10 Marks)
(d)	Draw a diagram to illustrate the vectors used in the Blinn-Phong specular reflection algorithm compared to basic Phong specular reflection.	(5 Marks)

Question 2: Texturing

(a)	Draw a diagram of a rectangular surface made from triangles, and provide texture coordinate values at each vertex, such that an image can be exactly mapped to the surface.	(5 Marks)
(b)	Briefly explain why aliasing happens in texture mapping.	(5 Marks)
(c)	Give two advantages of MIP-mapping, and state the cost of using it. Describe two variants to MIP-mapping and the problems they solve.	(10 Marks)
(d)	Explain how sky-boxes are created with cube-mapping. State how the box is created, how texture sampling is done, and how the box transforms relative to a virtual camera.	(5 Marks)

Question 3: Linear Algebra and Geometry

(a)	Give a series of equations or pseudo-code that determine a normal for a triangle, given its three vertex points.	(5 Marks)
(b)	Give the components of the result, *w*, for both of: i) `vec4 w = T * R * vec4 (0.0, 0.0, -1.0, 1.0);` ii) `vec4 w = R * T * vec4 (0.0, 0.0, -1.0, 1.0);` where *T* is a 4x4 translation by (1.0, 0.0, 0.0) matrix and *R* is a 4x4 Y-axis rotation by -90 degrees matrix. Matrices are in column-major notation.	(5 Marks)
(c)	Suppose you have an aircraft simulation. You know the aircraft's • world position; • forward vector; • up vector. Write a pseudo-code function to determine if a given world position is ahead or behind, and to the left or the right of the aircraft. You may use a diagram to explain your method.	(10 Marks)
(d)	Give two advantages or useful properties of using a unit quaternion (versor) to calculate an orientation or rotation, and state the variables or vectors used as inputs to creation of the versor.	(5 Marks)

Question 4: Virtual Cameras

(a)	Briefly define both of the following coordinate spaces: i. world space; ii. eye space.	(5 Marks)
(b)	Draw a diagram showing the variables and vectors used as inputs to a `lookAt()` style function.	(5 Marks)
(c)	Write pseudo-code or C to show how you would create and update virtual camera matrices for a camera that continuously orbits (rotates around) a stationary object, and has the following properties: • has no user controls; • is always fixated on (pointing towards) a point [-10.0, 0.0, 5.0]; • is always 7.0 units away from the above point; • has a viewport size of 1920x1080 pixels. Any other variables required may be defined by you. It is not necessary to call any OpenGL functions. You may approximate calls to any mathematical or time functions.	(10 Marks)
(d)	Write a shader in pseudo-code that will render any given geometry white, except if the geometry falls within a 0.5x0.5x0.5 cube in the middle of view in normalised device coordinates. It shall then be rendered red. You may assume that you have uniforms for perspective, view, and model matrices, and a position input in a local coordinate space.	(5 Marks)

Question 5: Hardware Pipeline and Shaders

(a)	Describe a shader-based method for visualising texture coordinate values and how you would use the visualisation to check the values.	(5 Marks)
(b)	State two reasons why storing geometric data in the graphics hardware's onboard memory is beneficial to rendering performance.	(5 Marks)
(c)	With reference to stages in the hardware pipeline, explain why built-in optimisations such as near and far-plane clipping do not prevent all computation of geometry that is out of view. Provide pseudo-code or mathematics to suggest an additional test that you can write to remove out-of-shot geometry entirely. Also explain how you would determine if large or complex meshes are out-of-shot.	(10 Marks)
(d)	Briefly explain the function of the tessellation stages in the modern hardware pipeline.	(5 Marks)

Question 6: Miscellaneous

(a)	Briefly explain how vertex and fragment shaders take advantage of the multi-processor architecture of graphics hardware.	(5 Marks)
(b)	Explain how back-face culling works.	(5 Marks)
(c)	Give a pseudo-code function to update the hierarchical animation of a car with four rotating wheels. The car body and wheels are two separate meshes. Your function inputs are: • Car heading in degrees; • Car world position; • Current wheel angle in degrees (same for all wheels). Position offsets for each wheel are: • [-0.5, 0.0, -1.0] (front-left) • [0.5, 0.0, -1.0] (front-right) • [-0.5, 0.0, 1.0] (rear-left) • [0.5, 0.0, 1.0] (rear-right) You function must output: • car body model matrix; • four car wheel model matrices; • wheel matrices must combine local and parent transformations. You do not need to steer the front wheels left or right. You may assume any mathematical functions. You do not need to make any OpenGL function calls.	(10 marks)
(d)	Give 2 potential performance bottlenecks in rendering your 3d software. Suggest a possible solution for each bottleneck.	(5 Marks)