

Monash University

Clayton's School of Information Technology

CSE3313 Computer Graphics Sample Exam Questions 2007

INSTRUCTIONS:

Answer all questions.

Spend approximately 1 minute per mark.

Question 1

30 Marks Total

- (a) What are the two main types of text supported in OpenGL?
[2 marks]
- (b) List the differences between these two text types and the advantages and disadvantages of using each in a graphics application.
[4 marks]
- (c) What kind of geometry is best described using the following OpenGL primitive types? Be specific about the relation between vertices and edges in your answer.
(i) `GL_TRIANGLE_STRIP`
(ii) `GL_QUAD_STRIP`
(iii) `GL_TRIANGLE_FAN`
[6 marks]
- (d) What is the best way to deal with concave polygons in OpenGL?
[4 marks]
- (e) What are the two main matrix modes used in OpenGL and how do they relate to the various coordinate spaces in the OpenGL pipeline?
[4 marks]
- (f) In the case of 2D orthographic projections, what relationship must hold between the display window, viewport and view volume in order to avoid aspect ratio mismatch? You may wish to use a diagram to illustrate your answer.
[5 marks]
- (g) What is the difference between pixel aspect ratio and display aspect ratio? How are the two related?
[5 marks]

Question 2**25 Marks Total**

- (a) Briefly explain the difference between a raster and a vector display.
[3 marks]
- (b) How many intensities can a 10 bit frame buffer display?
[3 marks]
- (c) List two different systems for describing colour. Why is there a need for more than one system of describing colour?
[3 marks]
- (d) What is meant by the term “double buffering” and for what purpose is it used?
[3 marks]
- (e) What is the difference between a frame buffer and a z-buffer?
[4 marks]
- (f) Explain how Bresenham’s algorithm determines how to draw pixels based on the gradient of the line. Use a diagram to assist with your explanation.
[6 marks]
- (g) OpenGL uses the concept of *attributes*. Explain what these are and give three examples of attributes.
[3 marks]

Question 3**35 Marks Total**

- (a) Many transforms in computer graphics are *affine*.
(i) What are the key properties of an affine transform?
(ii) Why are affine transforms so important when transforming polygonal data?
[4 marks]
- (b) Transforms are often carried out using a *homogeneous* co-ordinate representation. Why is this representation used?
[2 marks]
- (c) 2D viewing in OpenGL uses an orthographic view volume mapped to a viewport in a screen window. An OpenGL window has been defined with co-ordinates:

$$W = (wx_{min}, wx_{max}, wy_{min}, wy_{max})$$

viewport co-ordinates:

$$V = (vx_{min}, vx_{max}, vy_{min}, vy_{max})$$

Express as a single matrix the transform required to convert 2D points

from world co-ordinates to screen coordinates using the window and viewport method.

[12 marks]

- (d) Give the sequence of OpenGL commands to perform a rotation in the z axis of 45 degrees about the point (10.0, -5.0, 0.0)
[4 marks]
- (e) What are *display lists*? How are they useful in graphics programming?
[3 marks]
- (f) You have been asked to develop a simple real-time simulation of the solar system. This model only needs to show the Sun and the Earth. You may assume that geometric models have been defined representing the Sun and the Earth using display lists. Each object is defined relative to its own local coordinate system with the centre of the object at the origin.

Write a C function

`showSolarSystem(int dayOfTheYear, int hourOfTheDay)`
that draws the Sun and Earth correctly using OpenGL. For this function you may assume:

- the Earth completes an orbit of the Sun in exactly 365 days;
- the input parameter `dayOfTheYear` varies from 1 to 365;
- the input parameter `hourOfTheDay` varies from 0 to 23;
- the initial rotation of the Earth around the Sun at day 1 corresponds to 0 degrees;
- the initial orientation of the Earth in orbit corresponds to 0 degrees at hour 0;
- the distance between the centre of the Earth and Sun is exactly 1.49×10^8 units;
- the Earth completes a revolution about its own axis once every 24 hours;
- the Earth orbits the Sun in the $z = 0$ plane;
- the display lists for the Sun and Earth objects have ids defined as `SUN` and `EARTH` respectively, i.e. to call the Earth: `glCallList(EARTH);`
- the function does not have to do any animation — just call the objects with the appropriate transformations to place them with correct position and orientation.

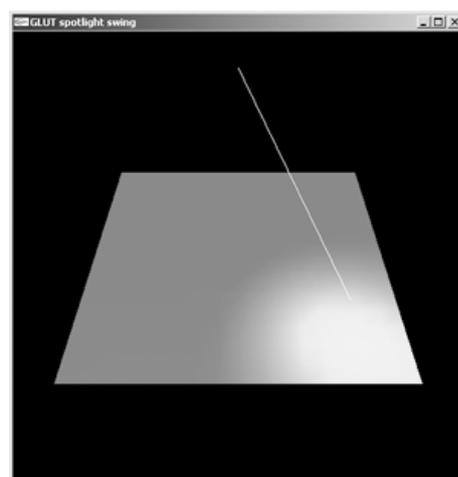
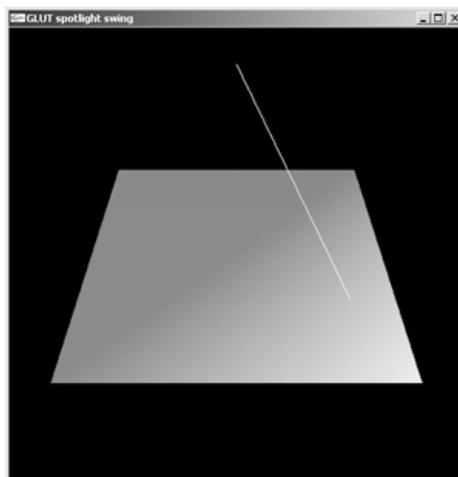
[10 marks]

Question 4

35 Marks Total

- (a) A polygon can be defined by a set of connected, co-planar vertices, $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$. What is the direction of the normal to the plane in which this polygon lies?
[5 marks]
- (b) Describe how the polygon normal can be used to help remove hidden surfaces. What viewing assumptions can reduce the cost of using this technique.
[3 marks]

- (c) In a simple computer graphics lighting model we assume the ambient component $\mathbf{I}_{\text{amb}} = k_a \mathbf{I}_a$.
- What is the purpose of the ambient component in a lighting model and what lighting situation does it approximate?
[3 marks]
- (d) How does the value of k_a affect the shading (in qualitative terms)?
[2 marks]
- (e) Is k_a a property of the light or the surface?
[2 marks]
- (f) Explain the difference between Gouraud and Phong interpolation for the simulation of smooth shading across faceted surfaces. Use a diagram to assist in your explanation.
[5 marks]
- (g) Explain how the Phong illumination model is extended in the global illumination ray-tracing algorithm to shade surfaces. You may wish to use a diagram to illustrate your answer.
[10 marks]
- (h) The two images below are both OpenGL scenes which contain one plane and one spot light each. In one of the images the plane is more highly tessellated than the other; no global effects, texture mapping or advanced shaders have been used. Explain the difference you see between the two images, why are they different?



[5 marks]

Question 5**40 Marks Total**

- (a) Briefly describe the *z-buffer* algorithm for hidden surface removal. [10 marks]
- (b) Does the algorithm operate in object space or image space? [2 marks]
- (c) In the case of the *z-buffer* algorithm, is shading performed before or after hidden surfaces are eliminated? [2 marks]
- (d) OpenGL uses a *z-buffer* to remove hidden surfaces. What factors affect the quality of hidden surface removal in OpenGL? [6 marks]
- (e) What is the *point normal* representation of a plane? How does it relate to the plane equation? [4 marks]
- (d) Show how the *z-buffer* algorithm makes use of coherence when calculating depth values as it scan converts a polygon. [16 marks]

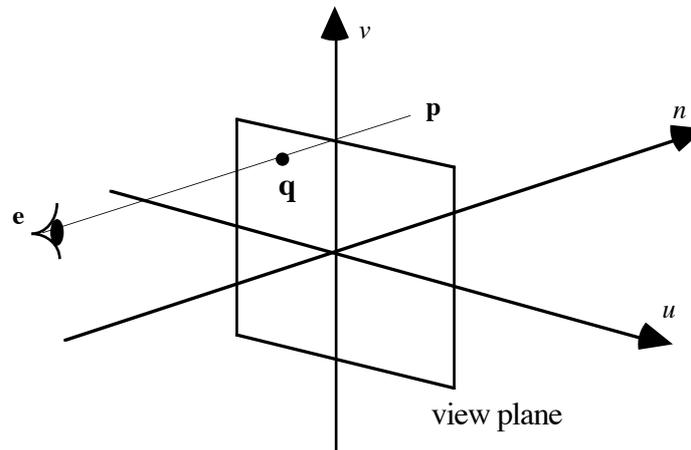
Question 6**[30 marks Total]**

A synthetic camera co-ordinate reference frame is given by a *view reference point* (**VRP**) a *view plane normal* (**VPN**) and a *view up vector* (**VUP**).

- (a) What are the data types of each of these three quantities? [3 marks]
- (b) Draw a diagram to show how these quantities describe the location and orientation of the synthetic camera. [4 marks]
- (c) The view up vector is normally resolved to be orthogonal to the view plane normal. Why? [3 marks]
- (d) The OpenGL call `gluLookAt` takes an *eye point*, an *at point* and an *up point*. Express VRP, VPN and VUP in terms of these three points. [6 marks]
- (e) Give the code necessary to implement `gluLookAt`. You should only need to make calls to OpenGL functions with the `gl` prefix (not `glu` or `glut`). [14 marks]

Question 6**25 marks Total**

Consider a left-handed view co-ordinate space (UVN) with the view plane at $n = 0$. An eye point (centre of projection), $\mathbf{e} = (e_u, e_v, e_n)$, lies behind the view plane (but not necessarily on the n axis):



- (a) Given an arbitrary point $\mathbf{p} = (p_u, p_v, p_n)$ which lies in front of the view plane ($p_n > 0$), what are the co-ordinates of \mathbf{q} , the point where the line \mathbf{ep} intersects the view plane? [8 marks]
- (b) If we now assume that \mathbf{e} lies on the n axis and is behind the view plane (therefore: $\mathbf{e} = (0, 0, e_n)$, $e_n < 0$), simplify the expression for \mathbf{q} obtained in part (a). [2 marks]
- (c) Using your result from part (b) express the perspective transform as a homogeneous transformation matrix given that \mathbf{e} lies on the n axis. [6 marks]
- (d) For a fixed imaging window on the view plane, what effect will changing the value of e_n have on the area projected onto the plane? [3 marks]
- (e) What is meant by the term ‘perspective division’? Show how it is applied using homogeneous co-ordinates. [3 marks]
- (f) Is clipping done before or after the perspective division? Why? [3 marks]

— End of Paper —