Royal Melbourne Institute of Technology
Department of Computer Science
CS541/493 Interactive 3D Graphics

Semester 1 2000

Date: Mon. 26th June
Duration: 3 hours
Time: 9:30am to 12:30pm
Number of Pages: 4

Instructions to Candidates:

The exam accounts for 50% of the total marks for the subject.
Candidates should attempt all questions 1 to 8.
Marks for each question are shown. Total marks for the exam is 100.

Question 1

(a) Each of the diagrams below is of a right-handed, three-dimensional coordinate system with one axis missing. Draw each coordinate system in your exam booklet, showing all three axes — including the missing axis. Label the missing axis and state whether it points into or out of the page.

(b) A triangle \( T_1 \) has the vertices \( A(2, 5, -4) \), \( B(6, 4, -5) \) and \( C(5, 7, -6) \). Work out the plane equation of the plane in which \( T_1 \) lies.

(c) A different triangle \( T_2 \) has the three vertices \( D(2, 7, -4) \), \( E(3, 3, -5) \) and \( F(7, 5, -6) \). Work out the plane equation of the plane in which \( T_2 \) lies.

(d) Which triangle from (b) and (c) has the largest \( z \) value at \( x = 4, y = 6 \)? Show your working.

(e) Assume a viewer is positioned at \( (4, 6, 1000) \) looking towards \( (4, 6, -1000) \). Which triangle from (b) and (c) would they see at the pixel \( (4, 6) \), assuming the viewplane lies in the \( xy \)-coordinate plane?

(f) If the hidden surface problem was to be solved by back-to-front drawing of polygons, in which order would \( T_1 \) and \( T_2 \) have to be drawn to get a correct hidden surface solution for the viewing arrangement of (d)? Justify your answer.

\[ 2 + 3 + 3 + 2 + 1 + 4 = 15 \text{ marks} \]
Question 2

(a) Three-dimensional computer graphics makes extensive use of $4 \times 4$ transformation matrices, rather than $3 \times 3$ transformation matrices. Explain why.

(b) Give the transformation steps and the combined transformation matrix (CTM) which transforms the object shown below on the left into the object shown below on the right.

(c) Write an OpenGL code fragment which draws the object on the right. Assume a function `displayObject()` which draws the object as shown on the left is available.

\[4+7+4 = 15 \text{ marks}\]

Question 3

(a) The scan-line algorithm for efficient rasterization of concave polygons uses an edge table and an active edge table. Give the algorithm, explain what these “tables” are and explain how they are used to realise an efficient algorithm. Include in your explanation a discussion of coherence and incremental arithmetic.

(b) Draw diagrams showing the state of the edge table and the active edge table in the algorithm you gave as answer to (a) at the start of scan lines $y = 0$, $y = 5$ and $y = 6$ for the polygon given below. Indicate the pixels which will be intensified on each of those scan lines.

(c) How may the efficiency of the polygon scan conversion algorithm be improved if only convex polygons are handled?

\[5+5+3 = 13 \text{ marks}\]

Question 4

(a) Explain the OpenGL transformation process used to produce a desired scene for viewing. Use a camera analogy and diagrams in your answer.

(b) Explain what viewing and modelling transformations are, including the difference between them, and discuss how they are treated in OpenGL. Use a diagram in your answer.

\[5+5 = 10 \text{ marks}\]
**Question 5**

(a) Hidden surface algorithms may be classified as *image space* or *object space*. Explain the classification and give at least one example of each type.

(b) BSP trees may be used to solve the hidden surface problem. Describe how the BSP tree hidden surface algorithm works and state and explain why it is an image space or object space algorithm.

(c) Using a diagram show the state of a BSP tree as each of the line segments shown below are inserted in the order A, B, C, D, E, F.

(d) Assuming the BSP tree in your answer to the previous question is to be used to solve the hidden surface problem, give the order in which the line segments and partial line segments, if any, will be drawn for a viewer positioned at the origin looking towards the point (10, 10).

$4 + 5 + 5 + 4 = 18$ marks

**Question 6**

OpenGL supports a range of rendering modes. In the diagram below a sphere has been rendered in five different ways.

![Rendered spheres](image)

(a) Describe the most important features of each rendering, explain what the settings or modes of the OpenGL state machine need to be to produce it and give the OpenGL command which applies to each setting or mode required.

(b) Discuss the effect of the different ways of rendering shown above on rendering speed in an animation.

$6 + 3 = 9$ marks
Question 7

(a) Explain what you understand of the following lighting formula. Use diagrams to clarify your answer.

\[ I_\lambda = I_{\kappa \lambda} + I_{\delta \kappa} k_a + \sum_{i} f_{\text{att}} \left[ I_{d\kappa} k_d (N \cdot L) + I_{e\kappa} k_e (R \cdot V) \right] \]

(b) Explain what is meant by, and the relationship between, the terms: Gouraud shading, Phong shading, flat shading and smooth shading.

\[ (6+4 = 10 \text{ marks}) \]

Question 8

A programmer writing a program to do facial animation has started with a simple head model consisting of only the neck, head and two eyes. The head is able to "shake" left and right by rotating about its vertical axis, and to "nod" forward and back by rotating about the point where it connects to the neck. The eyes are fixed and move with the head.

The model consists of a sphere for the neck, an ellipsoid for the head and cylinders for the eyes. The dimensions and positioning of the model are shown below.

![Diagram of head model](image)

Given a function `solidSphere(float radius)` which draws a sphere centred on the origin, write a function

```c
void display(void)
```

which uses OpenGL modelling transformations and matrix stack manipulation to draw the model. The "nod" and "shake" rotations are specified by two global variables `nodRotation` and `shakeRotation`. The head should be upright and facing forward when both the `nodRotation` and `shakeRotation` are 0.0. Assume all viewing parameters, window initialisation, etc., have been correctly set up.

\[ (10 \text{ marks}) \]

THE END