UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1998

BEng Honours Degree in Computing Part III
BEng Honours Degree in Information Systems Engineering Part III
MEng Honours Degree in Information Systems Engineering Part III
BSc Honours Degree in Mathematics and Computer Science Part III
MSci Honours Degree in Mathematics and Computer Science Part III
MEng Honours Degree in Electrical and Electronic Engineering Part IV
MSc Degree in Computing Science
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the
Diploma of Membership of Imperial College
Associateship of the City and Guilds of London Institute
Associateship of the Royal College of Science

PAPER 3.17 / I3.16 / E4.32

INTERACTIVE GRAPHICS
Wednesday, April 29th 1998, 10.00 - 12.00

Answer THREE questions

For admin. only: paper contains 5 questions

1. Algorithms

a. Describe the general idea of space division algorithms and discuss how they help making graphics processing tasks more efficient in the two cases when there are a large number of small objects with small facets to render and when there are a few objects with large planar faces in the graphics scene.

b. Describe both the pixel based and the scan line based z-buffer algorithms. If you had to implement one of these algorithms in software for a specific task and assuming that you did not have the time to code both to compare their execution times, how would you decide which one to code for a given graphics processing task?

c. Scan line algorithms are known to be very efficient. What aspects of scan line algorithms contribute to their efficiency?

The three parts carry respectively 30%, 40%, 30% of the marks.

2. Shading

Parallel (orthonormal) projection in the z direction is used for rectangular faces in 3D with the planes of the faces always parallel with the x axis; therefore, when projected, they remain rectangular. The projected lower left hand corner point ***C0****[X0,Y0]* of such a face, and its projected apparent height *(H)* and width *(W)* are all given by integer pixel numbers.

a. Show the code or pseudo code which uses Gouraud shading to render this rectangular area when the intensities at the corners are *i1, i2, i3,* and *i4* starting from the lower left corner and proceeding in a clock-wise direction. The intensities, which are floating point numbers, directly translate to the proper colour number when the floating-point to integer conversion is made.

b. Describe the procedure (do not show code or pseudo code) which uses Phong shading to render the same rectangle when the view point and a single point light source are at the origin and the four corners of the rectangle in 3D are given by position vectors ***P1, P2, P3,*** and ***P4***.

The two parts carry equal marks.

3. Vectors, Matrices and Transformations

a. Show the transformation matrix which rotates an object by ϕ degrees with respect to an axis that goes through the centre point of the object ***C0****[x0,y0,z0]* and points in the direction [1,1,0]. The rotation is clock wise when it is viewed from the centre of the object in the direction of the axis, i.e. [1,1,0]. Show the answer as a product of matrices, showing all the element values but not multiplying them out.

b. Two planes, which do not cross the origin, are given by two position vectors ***P1****[x1,y1,z1]* and ***P2****[x2,y2,z2]*. Each plane goes through the point specified by its respective position vector and its normal is parallel to the same vector. Derive an expression for the line of intersection between these two planes and calculate two points on this line for the case when ***P1*** = [3.0,4.0,2.0] and ***P2*** =[-3.0, -2.0, 2.0].

The two parts carry equal marks.

4. Ray Tracing

a. Derive a formula to determine whether a ray which is parallel with the z axis, and goes through the pixel located at ***Ppix****[xpix,ypix,f]* intersects a sphere having a centre at ***C0****[x0,y0,z0]* and radius *R*. All values are given in user (floating point) coordinates.

b. Derive a formula for the intersection point of the same ray as in part a. and a square shaped planar facet with its centre at ***SQ****[xsq0, ysq0, zsq0]* . The normal to the facet plane is direction vector $\boldsymbol{n}$ and the length of its side is w . Two of the sides of the square are parallel with the x axis.

c. Investigate whether using a bounding sphere for the square shaped facet in part b. could save you processing time if the number of visible facets in the graphics window is *NV* and the total number of facets is *NT*.

The three parts carry respectively 30% 40% 30% of the marks.

5. Raster Algorithms

We have a hardware based super fast line drawing procedure which works for the conditions: $XPixels>X2\geq X1\geq 0$; $YPixels>Y2\geq Y1\geq 0$; $(X2-X1) \geq (Y2-Y1)$ where **P1** = *[X1,Y1]* and **P2** = *[X2,Y2]* are the two end points of the line in pixel numbers in the image buffer *IBuf[0..XPixels-1][0..YPixels-1]*. The results are returned in a one dimensional array *Y[0..XPixels]*, which are the y pixel values on the line stored in the array elements *Y[0]* to *Y[X2-X1]*, and which must be set to the line colour when the x values run from *X1* to *X2*. The calling sequence is:

```
HardwareBresenham(int X1, int Y1, int X2, int Y2, int Y[])
```

There is no pre-condition testing in the routine.

a. Show how this procedure may be used to fill the image buffer correctly with a line which is within the frame buffer area: $0\leq X1,X2<XPixels$ and $0\leq Y1,Y2<YPixels$, and is in the first quadrant (i.e., X2>=X1 and Y2>=Y1). Show the required code or pseudo code and calling sequence of your more general line drawing routine which uses this procedure.

b. In the user's or World coordinate system a line is given by two 3D points ***P1****[x1,y1,z1]* and ***P2****[x2,y2,z2]*, six floating point components. Show the system parameters which you must know and the necessary calculations you must make to turn these into integer pixel coordinates. The view point is at the origin and the principle viewing axis is in the $+z$ direction.

c. Show the algorithm or code which sets the required pixel values to a given colour number for a circle with centre point ***C0****[X0,Y0]* and radius *R* in a *NxN* image buffer (the circle is inside the image buffer area, or: $0\leq X0\pm R, Y0\pm R<N$) using the error correction algorithm. The values are all given as integer pixel numbers.

The three parts carry respectively 40%, 20%, 40% of the marks.

End of Paper