CS3240

THE UNIVERSITY OF WARWICK

Third Year Examinations: Summer 2015

Computer Graphics

Time allowed: 3 hours.

Answer **FOUR** questions.

Read carefully the instructions on the answer book and make sure that the particulars required are entered on each answer book.

Use of approved calculators is permitted in this examination.

---

1. (a) Explain how the human visual system's capabilities are exploited by computer graphics. [4]

   (b) What are the standard components of a graphics system? Illustrate with a diagram the main stages of a 3D viewing pipeline. [7]

   (c) In OpenGL, what are matrix stacks? What do the following statements each do and which matrix stack would you normally use them on?

   i.
   ```
   gluOrtho2D(0, 1000, 0, 1000);
   ```

   ii.
   ```
   glTranslatei(500, 500, 0);
   glScalef(2.0f, 2.0f, 1.0f);
   ```

   [7]

   (d) Show how a rotation of θ about an arbitrary axis, $(a, b, c)^T$, through the origin, can be achieved by suitable rotations about the principal axes of θ_x and θ_y and θ_z respectively. Calculate coefficients of rotation matrices $R_x(\theta_x)$ and $R_y(\theta_y)$ and give a set of OpenGL statements which will perform $R(\theta)$, if the angles are given in degrees. [7]

Continued

2. (a) Giving examples, distinguish between diffuse and specular reflection. [4]

   (b) Describe and illustrate the Phong lighting model for a single point light source. Explain carefully the purpose of the parameters of the model. [7]

   (c) Calculate an expression for the Phong *shading* value at the origin, of a triangle with vertices:

   $$(\text{-}2, \text{-}2, 0) \quad (4, \text{-}2, 0) \quad (0, 2, 0),$$

   if the vertices have corresponding illumination and normal values:

   $$I_1, \mathbf{n}_1 \quad I_2, \mathbf{n}_2 \quad I_3, \mathbf{n}_3.$$

   [7]

   (d) Explain how texture mapping works, giving the necessary coordinate transformations required. [5]

   (e) An image needs to be mapped to a rectangle of size `width` by `height` in OpenGL. Give the missing statements in the following code fragment which specifies source and target coordinates:

```
glBegin(GL_QUADS);
    // missing statement 1
    glVertex2i(0, 0);
    // missing statement 2
    glVertex2i(width, 0);
    // missing statement 3
    glVertex2i(width, height);
    // missing statement 4
    glVertex2i(0, height);
glEnd();
```

   [2]

Continued

3. (a) Bresenham's line algorithm uses the decision parameter $P(k)$ for the k^{th} step of the method:

$$P(k) = 2\Delta y\ x(k) - 2\Delta x\ y(k) + K.$$

If the line goes between the end points (x_1, y_1) and (x_2, y_2):

i. Step-by-step, derive and expression for $P(k+1)$, and the constant K. [5]

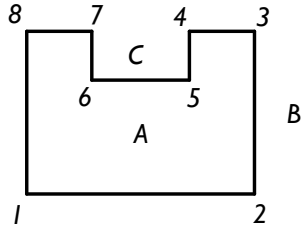
ii. What two values can $P(k+1)$ take and what does that mean for the algorithm? [3]

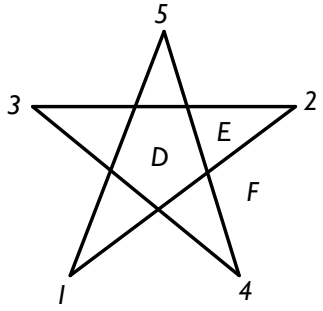
iii. What is the initial value of $P(k)$, and which values can be pre-calculated knowing only the endpoints of the line? [3]

iv. What conditions need to be imposed on the slope? How can arbitrary slopes be dealt with without sacrificing efficiency? [4]

(b) How could antialiasing be incorporated into Bresenham's line algorithm? [5]

(c) What are the winding-number values at locations A, B, C, D, E, and F for the following two polygons? (Vertices are numbered in the order they are drawn).





[5]

Continued

---

4. A 3D object, O, can be viewed from a position $\mathbf{p}$ by looking in direction $\mathbf{v}$.

   (a) Sketch the viewing geometry giving all parameters required to map the object on to a view plane. [5]

   (b) If a viewport on the view plane has size $W \times H$, what 3D and 2D transformations are required to project vertices of O ensuring that those inside the viewport are in the range $[0, 1]$? [6]

   (c) How can the edges of the object be efficiently clipped to the viewport using the Sutherland-Cohen method? Illustrate your answer with three typical cases. [7]

   (d) Write down the set of inequalities for the line $(x_1, y_1) \rightarrow (x_2, y_2)$ which are necessary to set up the Liang-Barsky clipping algorithm on a normalised viewport. State the parameter values for intersections of the line with the four edges of the viewport. [7]

---

5. (a) How can a Z-buffer be used to correctly deal with hidden surfaces? What instructions are required to use depth buffering in OpenGL? [3]

   (b) How can multiple transparent and opaque surfaces be properly blended using Z-buffers and alpha blending? [5]

   (c) What is back-face culling and how is it implemented? [4]

   (d) How can ground-plane shadows be generated of a polygon mesh consisting of 3D points $\mathbf{P}_i$, for a distant light source at world coordinate $\mathbf{L}$, assuming that the ground plane is at $z = 0$? [6]

   (e) What are shadow volumes and how can they be used together with stencil buffers to produce realtime shadows? [7]

---

6. Write short descriptive and illustrated explanations on the following:

   (a) Environment and bump mapping. [8]

   (b) Bresenham's mid-point algorithm for efficient scan conversion of circles, showing how the decision variables at each step are derived and what the initial conditions need to be. [8]

   (c) Scan-line fill algorithms for polygon, depth map generation and interpolated shading calculation. [9]

7. (a) Work out the Bezier polynomials for a cubic spline. [5]

   (b) Derive a Bezier matrix, $\mathbf{B}$, for a cubic spline satisfying the blending formula:

   $$\mathbf{q}(u) = \mathbf{UBb}$$

   making the definitions of $\mathbf{U}$ and $\mathbf{b}$ clear. [7]

   (c) Derive the conditions under which 1st and 2nd order continuity can be achieved between successive knots of the spline designed above. [7]

   (d) Given the forward-differing approximation:

   $$\Delta x(u) = x(u + \delta) - x(u),$$

   find an expression for $\Delta x(u)$ if

   $$x(u) = a_3x^3 + a_2x^2 + a_1x + a_0.$$

   Explain why this result is helpful in reducing the calculations required to draw splines. How many operations are required to calculated one step, δ, forward when drawing a 2D cubic spline using forward-differencing? [6]

End