Name:

Final examination

Computer Graphics 1 (15-462)

Grade value: 22%. Total marks: 160

Duration: 3 hours

Answer all questions for full credit

Question 1: Fill algorithms (20)

(a) What is the difference between flood filling and boundary filling? (2)

(b) What is the simplest implementation of filling? (1)

(c) Why is this approach unsatisfactory? (1)

(d) What is the difference between a 4 connected and an 8 connected fill algorithm? Illustrate with a diagram. (2)

(e) What implication is there to using either 4 or 8 connected fill? Consider what happens when the two algorithms reach a sloping line one pixel wide. (3)

(f) Span based filling is an alternative to the recursive fill algorithm. What is a span in flood filling terms and in boundary filling terms? (2)

(g) What data structure is used to keep track of the spans in this algorithm and how does this improve on the recursive algorithm? (2)

(h) Give an algorithm for a simple span based fill algorithm. Include the initialization step and how spans are added to and removed from the stack. (5)

(i) How is the above algorithm made to give 4 connected or 8 connected fill? (2)

Question 2: Scan conversion (30)

(a) What is scan conversion? (1)

(b) Write out real code for the simple Differential Digital Analyzer (DDA) algorithm for scan conversion of lines. (5)

(c) What assumptions does this code make about the ordering of the points and the slope of the line? (2)

(d) What can be done in the cases where these conditions are not true? (2)

(e) With a sketch illustrate which pixels are drawn for an example line with slope greater than 0 and less than 1. Do not make precise calculations, simply illustrate a general appearance. (2)

(f) It is possible to do the scan conversion calculations using entirely integer operations. What is the name of this algorithm? (1)

(g) The algorithm is based on the implicit function of a line. Write down the implicit function of a line and define and explain the components of the equation. (4)

(h) How is the implicit function useful in determining which pixel to draw next? (2)

(i) Calculating the value of the implicit function still requires the calculation of a dot product - how is this avoided? Show your working. (4)

(j) The code to do this integer line drawing is:

```
void draw_line ( int x0, int y0, int x1, int y1 ) {

    int x, y = y0;

    int dx = 2*(x1-x0), dy = 2*(y1-y0);

    int dydx = dy - dx, d = dy - dx/2;

    for ( x = x0 ; x <= x1 ; x++ ) {

        draw_pixel ( x, y );

        if ( d < 0 ) { d += dy; }

        else { y += 1; d += dydx; }

    }

}
```

Annotate this code to explain what each variable represents and what each command does. (6)

(k) How many integer operations are required per pixel in this code? (1)

Question 3: Visibility algorithms (20)

(a) In the display of 3D scenes what is the visibility problem? (1)

(b) Painter's algorithm solves the visibility problem by sorting the scene objects by depth. Write some very simple pseudo-code which implements Painter's algorithm, giving the range of each of the loops used. (5)

(c) The main loop of this function is extremely simple - explain why it solves the visibilty problem. (1)

(d) Where is the complexity in this algorithm which makes it difficult to implement? Mention the most significant problem. (2)

(e) The z-buffer algorithm also solves the visibility problem. Explain its operation with pseudo-code, including its initialization step and the ranges of the loops. (5)

(f) Where might this algorithm have problems? (1)

(g) The z-buffer algorithm can also be used to compute shadows in 3D scenes without having to compute ray tracing. Explain how the algorithm can be altered to achieve this. (5)

Question 4: Viewing (20)

(a) What transformations are required to place a camera at any suitable position to view a scene? (2)

(b) What three (mainly common sense) considerations go into the positioning of the camera in a scene? (3)

(c) Describe a simple view specification scheme in which the three requirements that you listed above are met. (3)

(d) Describe how each of the above sections of the viewing scheme is implemented. Explain how each component of the transformations required can be calculated. Recall that the rotation matrix for an angle-axis rotation of θ about axis $\vec{v}$ is given by: $R = \vec{v}\vec{v}^\top + \cos\theta(I - \vec{v}\vec{v}^\top) + \sin\theta\vec{v}^*$ (6)

(e) When might this viewing scheme break down? Why? (2)

(f) Explain what each of the three components of the equation that gives the angle-axis rotation matrix (from part (d)) achieve. (4)

Question 5: Illumination (40)

(a) What is light and what gives it different colors? (2)

(b) Why do we encode light as only three numbers? (1)

(c) What would be two implications for television if it had to reproduce the whole spectrum of light? (2)

(d) What is the difference between additive and subtractive color? (2)

(e) What three components of illumination (not color) are used to calculate shading for an opaque surface? (3)

(f) What two simple attenuation effects can be added to these illumination calculations? Explain the physical reasons for using these effects and give a simple equation if possible. (5)

(g) If a surface is not totally opaque then transmission of light also occurs - what happens to the light rays if the refractive indices of the two media are different? (1)

(h) It is very difficult to compute the above illumination terms using physical models and so approximations are used. A commonly used illumination equation is from Phong:

$$I = I_a k_a + f_{att} I_p (k_d \cos\theta + k_s \cos^n \alpha)$$

Explain all the values and terms in the above equation. (10)

(i) The above equation refers to a single light source - how is the equation extended for multiple light sources? (2)

(j) How is color implemented using the above equation? (1)

(k) The above equation still does not produce very realistic images - what do the images look like and why are they unrealistic? (1)

(l) Texture mapping can be used to improve images. There are several different ways of using it, name four of them. (4)

(m) Briefly explain how texture mapping is implemented. (4)

(n) One method of texture mapping generates the appearance of a rough surface on an actually planar surface. State how this is done and describe the anomaly that this approach produces. (2)

Question 6: Ray tracing (30)

(a) Explain how ray casting works. (3)

(b) Explain how recursive ray tracing extends ray casting. (2)

(c) Typically four different kinds of rays are used during recursive ray tracing. What are these four types of ray and what are their functions? (8)

(d) Three of the above rays cause more rays to be generated when they hit a surface, which of the rays doesn't? Explain why. (2)

(e) One of the simplest ray intersection calculations is that for a sphere - solve this intersection problem. Show all your working and give the solution in its simplest possible form. Recall that one of the solutions of a quadratic equation $ax^2 + bx + c = 0$ is $x = \frac{-b+\sqrt{b^2-4ac}}{2a}$. (5)

(f) There are five possible outcomes for the values of t (in terms of the number of solutions, and the signs of the solutions), list them and explain the physical significance of each. (10)