

Chapter 7

1. What does the value at each pixel in a bump map correspond to? How is this data used in rendering?
2. Describe environment maps, and explain the difference between the use of cube maps and spherical maps to implement them.
3. Explain the problem of rendering translucent objects using the z-buffer algorithm, and describe how the algorithm can be adapted to deal with this problem (without sorting the polygons).

Chapter 8

1. What are the two main types of algorithms for hidden surface removal? Briefly describe the *z-buffer algorithm* and *Painter's algorithm*. To what main type does each algorithm belong?
2. In the case of the z-buffer algorithm for hidden surface removal, is shading performed before or after hidden surfaces are eliminated? Explain.
3. The simplest scan-conversion algorithm for line segments has become known as the DDA algorithm. Give the algorithm in pseudo-code form and explain briefly how it is derived.
4. Bresenham derived a line-rasterization algorithm that avoids floating-point arithmetic. Describe this standard approach used in hardware and software rasterizers.
5. Bresenham's line-rasterization algorithm has become the standard approach used in hardware and software rasterizers as opposed to the more simpler DDA algorithm. Why is this so?
6. Describe, with the use of diagrams, the Liang-Barsky clipping algorithm.
7. Describe, with the use of diagrams, the Cohen-Suntherland clipping algorithm.
8. Briefly describe the scan-line algorithm for scan conversion and filling of polygons.
9. Show how one can decide if a point is inside or outside a complex 2D polygon.

- 4 Explain what is meant by texture aliasing. Explain how *point sampling* and *linear filtering* help to solve this problem.
- 5 A fairly simple and cheap (computationally) way of obtaining realistic-looking images is to use texture maps. What are texture maps? How are they used and what are the difficulties facing the implementer of a graphics package when the map is to be applied on an image?
- 6 In OpenGL, what steps are needed in order to apply texture on a polygon.
- 7 Image compositing (or blending) is used to blend colour shades from several objects. Briefly describe how one can accomplish image compositing in OpenGL (from the application programmer's point of view) and give an example of when it is used.

General pipeline questions

- 1 In order to show primitives described by vertices, the graphics system must perform the following operations: transformation, rasterization (or scan conversion), projection, and clipping. In what order are these operations generally performed (in OpenGL, for example)? Graphics systems often use pipelining for these operations? What does this mean?
- 2 Assume that we have defined a number of objects with polygons. Besides the transformation to 2D, what must be done before we can show an image on a computer screen?
- 3 The pipeline for the rendering of graphics scenes consists of the following steps:

(1) Vertices/Edges/Facets/Objects -> (2) Transformations -> (3) Viewing -> (4) Clipping/Projection -> (5) Hidden Surface Elimination -> (6) Rendering.

Describe briefly the purpose of each step (1)-(6) of the pipeline.