

## Tutorial 1

### For Week 3

---

Please attempt the following questions before you go to your tutorial classes. Some of the questions may be quite open-ended and some may be even ambiguous. In those cases, you are encouraged to make your own (reasonable) assumptions.

- (1) To be able to display realistic images, our display devices need to be able to produce every frequency in the visible light spectrum. True or false? Why? What are the advantages and disadvantages?
- (2) Each pixel in a frame-buffer has 8 bits for each of the R, G and B channels. How many different colors can each pixel represent? Is this enough? On some systems, each pixel has only 8 bits (for all R, G, and B combined). How would you allocate the bits to the R, G and B primaries?
- (3) Referring to Lecture 1 Slide 36. If an imaginary image plane is  $d$  unit distance *in front* of the pinhole camera, what are the coordinates of the projection (on the imaginary image plane) of the 3D point  $(x, y, z)$ ? If the camera's center of projection is not located at the origin, and the camera is pointed in an arbitrary orientation, the calculation of the projection becomes very messy. How would you make it less messy?
- (4) Why do we need a primitive assembly stage in the rendering pipeline architecture?
- (5) What does the rasterization stage (rasterizer) do in the rendering pipeline architecture? Describe what it does to a triangle that is supposed to be filled, and the three vertices have different color. Assume smooth shading is turned on.

(6) What is a GLUT display callback function? Give example events for which the display callback function should be called.

(7) Which of the two following program fragments is more efficient? Why? Can the same optimization be done for the case of GL\_POLYGON?

<b>A</b>	<b>B</b>
<pre>double v[3*N][3]; ... for ( int i = 0; i &lt; 3*N; i+=3 ) {     glBegin(GL_TRIANGLE);     glVertex3dv( v[i] );     glVertex3dv( v[i+1] );     glVertex3dv( v[i+2] );     glEnd(); }</pre>	<pre>double v[3*N][3]; ... glBegin(GL_TRIANGLE);     for ( int i = 0; i &lt; 3*N; i+=3 )     {         glVertex3dv( v[i] );         glVertex3dv( v[i+1] );         glVertex3dv( v[i+2] );     } glEnd();</pre>

(8) What is hidden-surface removal? When is it not necessary?

(9) Devise a test to check whether a polygon in 3D space is planar.

(10) Devise a test to check whether a polygon on the  $x$ - $y$  plane is convex.