CE325: 3D Computer Graphics
Problems

Adrian F. Clark

Multiple-choice questions

1. What is the correct name for the transformation that converts between an *object coordinate system* and a *world coordinate system*?

   (a) an object transformation
   (b) a world transformation
   (c) an instance transformation
   (d) a class transformation

2. Which of the following is *not* a property of a transformation?

   (a) intersecting lines remain intersecting
   (b) the length of lines is preserved
   (c) points on a straight line remain in a straight line
   (d) parallel lines remain parallel

3. If the transformation matrices $\mathbf{S}$ represent scaling and $\mathbf{T}$ translation, what is the result of applying a translation followed by a scaling to the point $\mathbf{P}$?

   (a) $\mathbf{STP}$
   (b) $\mathbf{TSP}$
   (c) $\mathbf{SP} + \mathbf{T}$
   (d) $\mathbf{S}(\mathbf{P} + \mathbf{T})$

4. What rotation does the transformation matrix

$$\begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

   represent?

   (a) a rotation through θ around the x-axis
   (b) a rotation through θ around the y-axis
   (c) a rotation through θ around the z-axis
   (d) a rotation through θ around the $-x$-axis

5. Take the unit square defined by the four points $(0,0), (1,0), (1,1), (0,1)$ and rotate it by $30°$ around the origin. Where do the four points end up? (Note that $\sin 30° = 0.500$ and $\cos 30° = 0.866$.)

   (a) $(0,0), (0.866, 0.500), (0.366, 1.366), (-0.500, 0.866)$

   (b) $(0,0), (0.500, 0.866), (1.366, 0.366), (0.866, -0.500)$

   (c) $(0,0), (0.866, 0.500), (0.366, 1.366), (-0.500, 1.366)$

   (d) $(0,0), (0.866, 0.500), (0.366, 1.366), (0.500, -0.866)$

6. What is the purpose of *homogeneous coordinates*?

   (a) to allow all coordinates to be represented homogeneously

   (b) to allow translation to be represented as a multiplication

   (c) to allow transformations to be combined by additions

   (d) to allow rotation and scaling to use the same matrices

7. Which of the following is *not* a property of perspective transformations?

   (a) parallel lines appear to converge in a *vanishing point*

   (b) object size reduces as distance from the centre of projection increases

   (c) foreshortening of lines depends on both orientation and distance

   (d) the shape of the object is preserved

8. What are the two matrices that together perform transformations in OpenGL?

   (a) VIEWMODEL and PROJECTION

   (b) PROJECTION and MODELVIEW

   (c) PROJECTION and PERSPECTIVE

   (d) VIEWMODEL and MODELVIEW

9. What is the purpose of the reshape callback in OpenGL?

   (a) to change the shape of the model

   (b) to change the shape of the viewport

   (c) to re-calculate the camera properties when the window is re-sized or re-shaped

   (d) to re-calculate the modelling transformation when the window is re-shaped

10. Which of the following describes Gouraud shading?

    (a) interpolation of the shading across a facet

    (b) interpolation of the shading across a facet and a model for highlights

    (c) interpolation of the normal across a facet

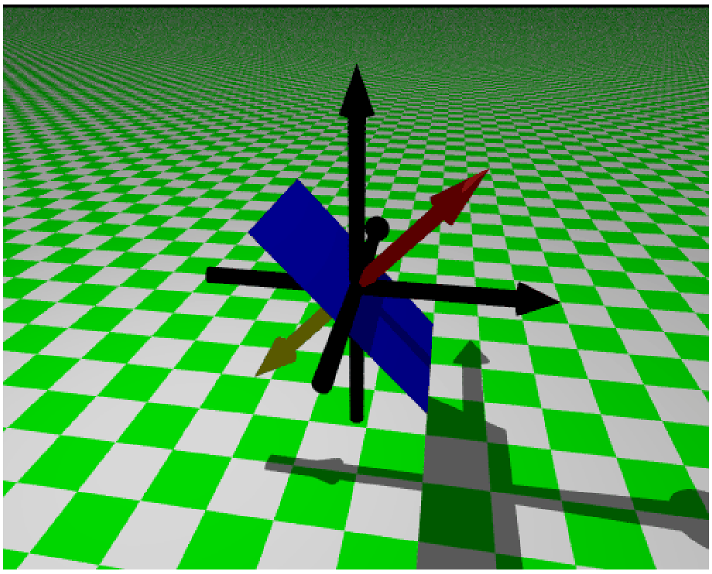
    (d) interpolation of the normal across a facet and a model for highlights

Short Questions

1. Obtain from first principles equations that describe how a point $\mathbf{P} = (x, y)'$ is rotated around the origin through an angle θ.
2. It is conventional for a display callback in OpenGL to have the same number of invocations of `glPush` and `glPop`. What do these do? Explain what will happen if there are more calls of `glPush` than `glPop`.
3. A polygon is defined by the four vertices $(0, 0), (1, 0), (1, 1), (0, 1)$. What is its normal vector?
4. What is the purpose of the near and far clipping planes in OpenGL?
5. What is *depth coherence*? Explain how it is able to speed up the painting of polygons in Z-buffer rendering.
6. What is *Mach banding* and how can its effects be reduced?
7. Explain how *texture mapping* can be used to make fairly simple 3D models appear much more realistic.
8. Describe in detail what feature is provided by OpenGL to avoid aliasing when texture-mapping.
9. What does it mean to say that a vector is *normalized*? Why is it important that normal vectors are normalized for performing shading in OpenGL?
10. Give an example of the use of 3D graphics in each of everyday life, science and medicine.

Long questions

1. Define a pyramid whose base is of side b and whose height is h as a set of polygons and as a polymesh. Compare the two in terms of the amount of data that needs to be stored.
2. Write a procedure to generate a sphere of radius R placed at the origin using OpenGL polygons. Suggest how the sphere may be generated with a different centre and any further information that would assist your procedure.
3. A scene contains a polygon defined by the points $(1, -1, -1)$, $(1, -1, 1)$, $(-1, 1, 1)$ and $(-1, 1, -1)$ and a visual representation of its upward-pointing and downward-pointing normal vectors.



(a) Calculate the normal vector to the polygon.

(b) Devise an OpenGL program to display the polygon and normal vector as wireframe objects.

(c) Extend the program to use solid, rather than wireframe, objects, lighting and hidden surface removal.

4. You have been asked to act as a consultant to Wivenhoe Software Technology (WST) because of your extensive experience of computer graphics. WST wish to produce a real-time animation to visualize some of their data. They have heard of various different approaches to 3D rendering but are unsure what approach to use or how to use the available software packages.

   (a) It has been decided that you will develop a visualisation program using the OpenGL libraries. WST's data are to be represented by a series of coloured boxes in the vicinity of the origin. The visualisation involves flying the camera continually in a circle around the x-axis, looking at the origin all the time. There is to be a 5° angular increment between frames.

   Write a program in pseudo-code that uses OpenGL calls to perform the required animation. It is not necessary to produce a syntactically-correct C program, but the required structure of an OpenGL program should be presented along with an indication of which OpenGL routines need to be invoked in which routines. It is also not necessary to write the body of a display routine.

   (b) The initial implementation of the complete visualisation program by WST works but flashes badly during the animation. What is the most likely cause of this and how can it be fixed?