Question 1

(a) \( \frac{60}{180} \pi = \frac{\pi}{3} \)

(b) \( \frac{4}{\sqrt{1^2+1^2+1^2}} \Rightarrow \frac{4}{\sqrt{3}} \Rightarrow \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} > \)

(c) Orthogonal, or zero

(d) Parallel, or zero

(e) Rasterisation of lines and circles

(f) In-order, back to front

(g) Right-handed

(h) 0

(i) Modelview

(j) glutPostRedisplay
Question 2

(a) (i) \( \vec{A} \cdot \vec{B} = 2 \times 0 + 2 \times 0 - 1 \times 4 = -4 \)

\[
\vec{A} \times \vec{B} = \begin{vmatrix}
i & j & k \\
2 & 2 & -1 \\
0 & 0 & 4 \\
\end{vmatrix}
\]

= i(8 - 0) + j(0 - 8) + k(0 - 0) = 8i - 8j

(ii) \(|\vec{A}| = \sqrt{2^2 + 2^2 + 1^2} = \sqrt{9} = 3 \)

\(|\vec{B}| = \sqrt{0^2 + 0^2 + 4^2} = \sqrt{16} = 4 \)

(iv) \(cos \theta = \frac{\vec{A} \cdot \vec{B}}{|A||B|} = \frac{-4}{3 \times 4} = -\frac{1}{3} \)

\(\theta = \cos^{-1}(-\frac{1}{3}) \)

(v) \(\frac{\vec{A}}{|A|} = \frac{2i+2j-k}{3} \)

\(\frac{\vec{B}}{|B|} = \frac{4k}{4} = k \)

(b) \(\vec{a} \vec{b} = \vec{b} - \vec{a} = (3, 2, -1) \)

\(\vec{a} \vec{c} = \vec{c} - \vec{a} = (0, 2, 2) \)

Find the normal to the plane:

\(\vec{N} = \vec{a} \vec{b} \times \vec{a} \vec{c} \)

\[
\vec{N} = \begin{vmatrix}
i & j & k \\
3 & 2 & -1 \\
0 & 2 & 2 \\
\end{vmatrix}
\]

= i(4 + 2) + j(0 - 6) + k(6 - 0) = 6i - 6j + 6k

\(\rightarrow i - j + k \)

General form of plane equation:

\(Ax + By + Cz + D = 0 \)

\(x - y + z + D = 0 \)

Find D by substituting \(\vec{a} \):

\(3 + 2 - 2 + D = 0 \)

\(D = -3 \)

Plane equation:

\(x - y + z - 3 = 0 \)
Question 3

(a) Homogeneous transformation matrices allow unification of rotation, scale and translation into a single 4 x 4 matrix. For efficacy, any sequence of transformations can be combined into a homogeneous transformation which is applied to geometry at a fixed cost.

(b) (i) Translate \((-1, -5, 0)\)
    Scale \(\frac{\sqrt{2}}{2}\) in \(x\) and \(y\)
    Rotate \(-45^\circ\) w.r.t \(z\) axis
    Translate \((-3, 0, 0)\)

(ii) \(T_1 = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\)
    \(S = \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & 0 & 0 \\ 0 & \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\)
    \(R = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\)

\(T_2 = \begin{bmatrix} 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\)

(iii) CTM
    \[ = T_2.R.S.T_1 \]
    \[ = \begin{bmatrix} 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & 0 & 0 \\ 0 & \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \]
    \[ = \begin{bmatrix} 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & 0 & 0 \\ 0 & \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \]
    \[ = \begin{bmatrix} 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & 0 & 0 \\ 0 & \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \]
    \[ = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & -3 \\ -\frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \]
    \[ = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & -6 \\ -\frac{1}{2} & \frac{1}{2} & 0 & -2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \]

(c) glTranslatef(-3,0,0);
glRotatef(-45,0,0,1);
glScalef(\(\frac{\sqrt{2}}{2}\),\(\frac{\sqrt{2}}{2}\),1);
glTranslatef(-1,-5,0);
displayObject();
Question 4

(a) (i) 

gColor3f(0.0,0.3,0.3);
gEnable(GL_LIGHTING);
gLightModeli(GL_LIGHT_MODEL_TWO_SIDE,1);
gPolygonMode(GL_FRONT_AND_BACK,GL_LINE);
gDisable(GL_CULL_FACE);
...
gEnable(GL_CULL_FACE);
gCullFace(GL_BACK);
...
gEnable(GL_CULL_FACE);
gCullFace(GL_FRONT);
...
gDisable(GL_CULL_FACE);
gPolygonMode(GL_FRONT_AND_BACK,GL_FILL);
gShadeModel(GL_FLAT);

(ii) In OpenGL drawing two objects to the same position causes a problem called *z-fighting*. This is due to the z-buffer having insufficient precision to distinguish two closely positioned surfaces at a particular pixel.

One approach is to disable depth testing and updates while back-face culling in GL_LINE polygon mode to draw the white wireframe over the green sphere.

Another approach is to use the OpenGL glPolygonOffset function to offset the z of each rasterised fragment while drawing the white wireframe.

(iii) Phong shading achieves per-pixel specular highlights by interpolating the normals across each scanline and recalculating the lighting equation for each pixel.

Graphics hardware with a programmable fragment shader can be configured to implement phong shading. An alternative is to texture-map specular highlights using environment mapping. Another alternative is to decrease the size of polygons until per-vertex lighting resembles per-lighting lighting.

(b) • Emissive

\[ I_e \]

where

\[ I_e \] is the intensity of the emitted light.

• Ambient

\[ I_a k_a \]

where

\[ I_a \] is the intensity of the ambient light and

\[ k_a \] is the coefficient of ambient reflection ranging between [0, 1].

• Diffuse
\[ I_p k_d (N.L) \]

where \( I_p \) is the diffuse intensity of the point light source,
\( k_d \) is the diffuse reflection coefficient,
\( N \) is the unit normal vector to the surface and
\( L \) is the unit light source direction vector.

- Specular

\[ I = I_s k_s (R.V)^n \]

\( I_s \) is the specular intensity of the point light source,
\( k_s \) is the specular reflection coefficient,
\( R \) is the unit direction of (specular) reflection and
\( V \) is the unit viewpoint direction.

- Attenuation is applied to the diffuse and specular components of each light source.
- The attenuated diffuse and specular components are summed over all \( n \) light sources.
Question 5

(a) (i)

(ii) Sketch a simple sequence of keyframe positions for an animated walk of a leg of the ATAT.

(iii) Sketch the parameter curves for the three leg joints with time on the horizontal axis and angle on the vertical axis. Use linear interpolation.

(iv) void displayLeg(double t)

   {
      glPushMatrix();
      glRotatef(hipAngle(t),0,0,1);
      drawUpperLeg();
      glPushMatrix();
      glTranslate(0,-2,0);
      glRotatef(kneeAngle(t),0,0,1);
      drawLowerLeg();
      glPushMatrix();
      glTranslate(0,-1,0);
      glRotatef(ankleAngle(t),0,0,1);
      drawFoot();
      glPopMatrix();
      glPopMatrix();
   }

Assumptions:
- The leg and foot geometry is in y axis with the pivot point on the origin.
- The length of the upper leg is 2 units.
- The length of the lower leg is 1 unit.

(v) double interpolate(double t1, double v1, double t2, double v2, double t)

   {
      return (t-t1)/(t2-t1)*(v2-v1)+v1;
   }
Question 6

(a) • Transformation
• Lighting
• Projection
• Clipping
• Rasterisation
• Z-Buffer Hidden Surface Removal

(b) Preprocessing arranges information into a form that can be efficiently used by subsequent processing stages. Construction of the ET (edge table) is a pre-processing step in the scan-line algorithm. Coherence is similarity of data between subsequent steps of an algorithm. The AET (active edge table) need not be completely recomputed for each step of the scan-line algorithm. It can be updated only as necessary.

Incremental arithmetic utilises deltas or gradients to reduce computations to additions or subtractions, rather than recomputing positions using division or multiplication. The scan-line algorithm uses incremental arithmetic to follow the edges in the AET.

(c) A 2D polygon can be classified as convex if all of the internal angles are less than 180°. This can be implemented as a cross product at each vertex and is suitable for real-time computer graphics. Another test for a cocave polygon is to find a line that crosses edges of a polygon more than two times. This approach is not suitable for real-time computer graphics since the choice of lines to test against the polygon is obvious.

(d) The outcodes utilised by the Cohen-Sutherland Line-Clipping algorithm are based on classifying and endpoint of the line against the four edges of the clipping rectangle. The outcode is four bits in length: The first bit is 1 iff the point is greater than \( y_{max} \), the second iff the point is less than \( y_{min} \), the third iff the point is greater than \( x_{max} \) and the last iff the point is less than \( x_{min} \).

\[
\begin{align*}
1001 & \quad 1000 & \quad 1010 \\
0001 & \quad 0000 & \quad 0010 \\
0101 & \quad 0100 & \quad 0110
\end{align*}
\]

When both outcodes are 0000, the line is trivially accepted.

When the bitwise AND of the outcodes is non-zero, the line is trivially rejected.

(e) The graphics hardware approach to the Visible Surface Problem is the z-buffer algorithm. Each fragment from the rasteriser is compared to the depth buffer. If the fragment is closer than the depth stored in the z-buffer the fragment is drawn and the z-buffer is updated. Otherwise the fragment is not drawn and the z-buffer is not updated.

The advantage of this approach is that it is simple and can be efficiently implemented in hardware. A disadvantage of the z-buffer approach is that transparency can not be correctly handled since the order of blending is important.

(f) (i)

(ii) \( D C E_2 A E_1 B \)

(iii) A BSP tree can be computationally expensive to rebuild due to the (often competing) needs for minimal splitting and maximal balancing. Therefore, BSP trees are most suitable for static (unchanging) scenes or shapes that can be built in a pre-processing step.

THE END