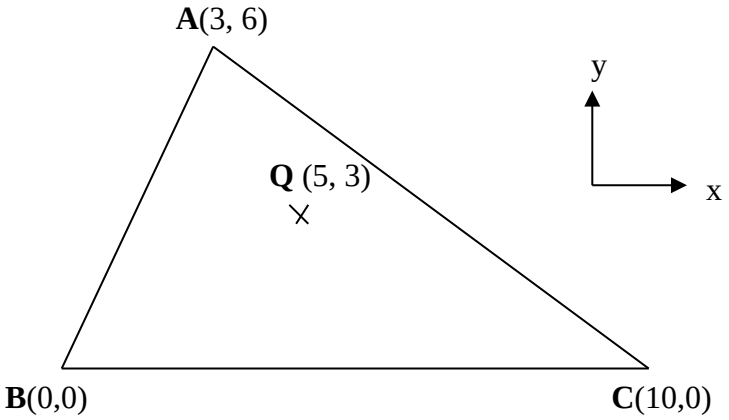
Tut 6

Qn 1

A quadrilateral mesh is composed of quadrilaterals. Each quadrilateral can be broken down into two triangles. Consider a triangle $\Delta\mathbf{ABC}$ to be displayed on the 2D image after perspective projection. It is needed to calculate the shade of the triangle.



The average surface normals at vertices **A**, **B**, **C** are

A = (0.6, 0, 0.8) **B** = (0.8, 0, 0.6) **C** = (0.6, 0.8, 0)

A far away light source has intensity $I_l = 1.0$, and the lighting direction vector is **L** = (0, 0, 1). Diffuse reflection occurs with $k_d = 0.7$. Scanning occurs from top to bottom, left to right.

a) Calculate the light intensity at image point **Q** using Gouraud shading.

b) Repeat a) using Phong shading.

Qn 2

Incremental form is a commonly used technique in CG to reduce the computation during rasterization.

a) Write the intensity of a point $\mathrm{P}(x, y)$ lying along the line segment $\overline{\mathbf{AB}}$ in incremental form.

b) How many multiplications/divisions and additions/subtractions are saved?

Qn 3

Calculate the intensity at the surface point with the following data:

Surface point (100,70,50) with unit normal vector $(\frac{1}{\sqrt{2}},0,\frac{1}{\sqrt{2}})$

A point light source at (100, 70, 1000) with intensity 2.0
Ambient light intensity 0.4
Ambient reflection coefficient 0.2
Diffuse reflection coefficient 0.6
Specular reflection coefficient 0.3
Specular reflection exponent 2
PRP of the viewer at (150, 120, 100)

OpenGL Mini-project Progress

Note: lighting and shading is NOT in the technical specification of the mini project. You can use default lighting.

Without lighting and shading definitions, OpenGL uses its default lighting and shading (e.g. a fixed colour for an object.) You are now ready to use your own lighting and realistic shading.

Within the function *your_function_name* in

glutDisplayFunc(your_function_name)

Delete all commands that set colours, e.g. glColor3f (0.0, 0.0, 1.0).

Add commands to define one or more light sources. Note that the light sources may also move.

Remember to enable the lighting system and also enable each light source.

Select a rendering mode (flat or Gouraud. Note that to use Phong you need OpenGL shading language, which is out of scope of this course.)

For each object, prefix it with the material properties you like it to have, i.e., add them before the object code.

Note 1: You cannot shade semi-transparent object at present. To do this, you need to know blending, which is in Techniques to Increase Realism notes. Semi-transparent object is not required in the Mini-Project.

Note 2: If you use quadrilateral mesh, you need to compute manually the normal for each vertex in the mesh.

You may like to read ahead to start program the shadow and/or texture.