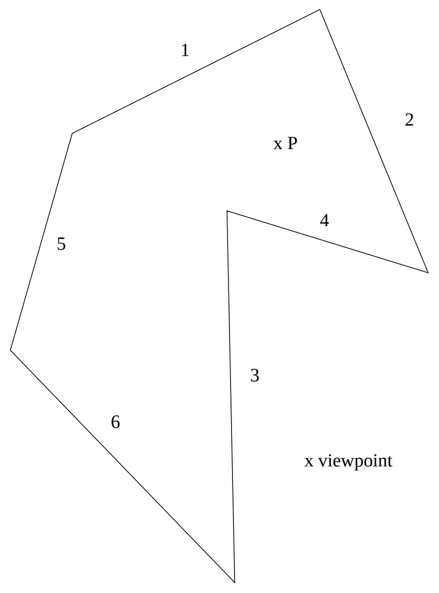
CS384G **Final Examination** **Fall 1997**

1. (27 points) Consider the following polygonal object in 2-dimensions.



- (a) Draw a BSP tree representing the object shown above, using partitioning lines defined by the line segments in the polygon in the order listed. Let left children be front halfspaces of the lines, and right children be back halfspaces, and assume all normals to the lines point to the outside of the polygon.

- (b) Sketch how the tree in part (a) partitions the plane. Label the regions of the plane and indicate which nodes in the tree correspond to these regions.

- (c) Give the list of line segments output in back to front order using your BSP tree from the viewing position shown.

- (d) Show the path through the your tree that would be taken in in determining whether the point P in the diagram is inside or outside the object.

2. (16 points) You have a ray-tracing program for which you do not have the source code, and you are trying to determine how it works internally. You are particularly interested in whether rays are always traced to a specific recursive depth, or whether they are terminated adaptively based on their contribution to the intensity of a pixel. Describe an experiment or set of experiments you could do to determine which is the case. Be specific about what scenes you would construct, how the lighting would be arranged, and what you would look for to make the determination.

3. (30 points) If any type of lighting is to be done to objects in a scene, one must be careful to do it correctly in the proper coordinate frame. Recall that objects are generally defined in their own model coordinate frames and are then mapped to world space and from there directly to viewing coordinates by the OpenGL modelview matrix. The perspective matrix then maps from viewing coordinates to homogeneous coordinates, and a division by w maps from homogeneous coordinates to 3D NDC space. Finally, a window to viewport transform performs a mapping to screen space. Lighting requires normals to surfaces (or vertices) to be transformed through these spaces as well as the vertices themselves.

- (a) We discussed the application of the Phong lighting model in world coordinates. If normals to objects are defined in the objects' model coordinate frames, how must they be transformed to world coordinates? Answer this by giving the matrices to apply to normal vectors for translation, scaling, and rotation and then specifying how to compose these.

- (b) Can the Phong model be applied correctly in viewing coordinates (the coordinate frame resulting from the application of the OpenGL modelview matrix)? Why?

- (c) Can the Phong model be applied correctly in 3D NDC space? Why?

- (d) Recall that if we do Phong shading, we interpolate the normal vectors at the vertices of a polygon to produce a normal at each pixel, and then we apply the Phong model to produce a color at each pixel. In which coordinate frame must this be done? Why?

- (e) Given your previous answers, describe how the Phong model is applied at each pixel in Phong shading. Specify what transforms, if any, are applied to the normals in this process.

4. (27 points) A bicubic Bezier surface patch has the form

$$\mathbf{P}(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3} P_{i,j}B_{i,j}(u,v) for 0 \leq u,v \leq 1$$

where

$$B_{i,j}(u,v) = b_i(u)b_j(v)$$

and for any parameter t

$$\begin{aligned} b_0 &= (1-t)^3 \\ b_1 &= 3t(1-t)^2 \\ b_2 &= 3t^2(1-t) \\ b_3 &= t^3 \end{aligned}$$

- (a) Show that the basis (blending) functions sum to one (i.e. that $\sum_{i=0}^{3} b_i(t) = 1$ for all t).

- (b) Give a formula for the normal vector to the patch at the point $\mathbf{P}(0.5, 0.5)$.

- (c) Can the midpoint subdivision scheme used for recursive display of Bezier *curves* be used for displaying patches? If so, describe how to use it. If not, explain why not.