CS 559: Computer Graphics

Homework 5

This homework must be done individually. Submission date is Tuesday, April 6, 2004, in class.

Question 1:

Consider the hidden surface removal algorithms presented in class. The table below has rows labeled with the algorithms, and columns labeled with statements about properties of visibility algorithms. The full form of all the statements is:

a. Requires knowledge about *all* of the polygons before it can begin drawing the scene.

b. Handles transparency without the user specifying polygons in any particular order (it figures out the order).

Fill in the table with T or F indicating whether each statement is true of false for each algorithm.

	All polygons reqd.	Easy transparency
Z-buffer		
A-buffer		
Depth Sorting		
Scanline HSR		
Warnock Subdivision		

Question 2:

The screen-space bounding box of an object is the smallest rectangle that can be drawn around it after projecting the object onto the screen. Its opposing corners are (x_{min}, y_{min}) and (x_{max}, y_{max}), which x_{min} is the minimum of the x coordinates of the polygon vertices, x_{max} is the maximum of the x coordinates of the polygon vertices, and similarly for y_{min} and y_{max}. Define the *quality* of an extent as the ratio of the number of pixels filled for the object to its extent's area.

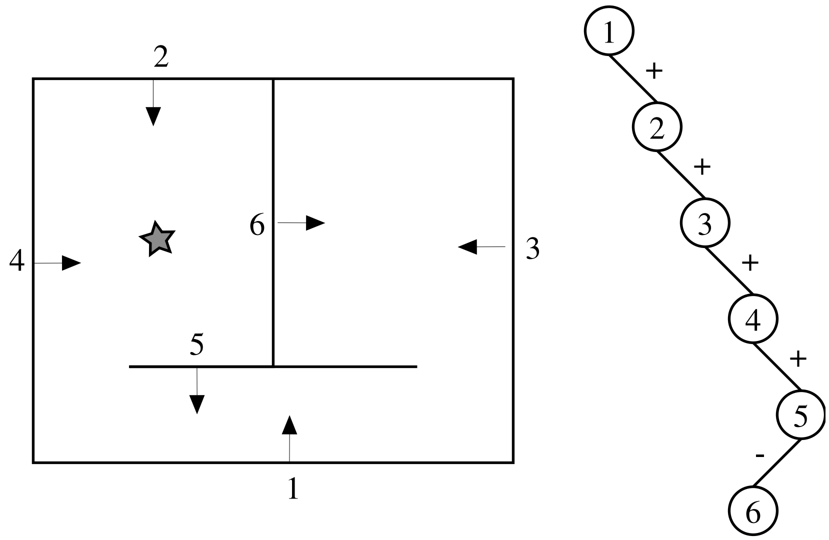
a. Which visibility algorithms (from the set in Question 1) have performance that is in some way related to the quality of the rectilinear extents of the rendered objects?

b. Give an example of an object and a viewing configuration that will yield high quality extents. For example, not valid as an answer, "an elephant viewed from above".

c. Give an example of an object that will typically yield very poor extents.

Question 3:

This question concerns BSP trees. One BSP tree is shown for the scene below.



a. Show the back-to-front rendering order for the BSP tree above if the viewer is located at the point marked with a star.

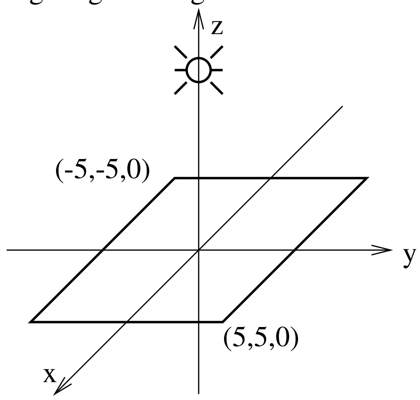
b. Build another BSP tree for the scene, this time using a *different* ordering that gives a BSP tree of lower height. You do not have to get the minimum height. Label nodes with the face numbers used to split.

c. Show the back-to-front rendering order for your BSP tree of part (b) if the viewer is located at the point marked with a star.

d. Which tree is more efficient for rendering the scene? Why?

Question 4:

This question explores the effect of the distant light source and distant viewer approximations. Assume you have a point light source with intensity 1 located along the positive z axis illuminating a white square with side length 10 located in the $x - y$ plane and centered at the origin, with diffuse reflectance 0.5 and specular reflectance 0.5 with no ambient reflectance. Assume the viewer is at the point (5,0,5) looking at the origin. For these questions use the standard lighting model given in class. The situation is pictured below.



a. Assume the light is at the point (0,0,10). What is the *diffuse* intensity at the center of the square?

b. Again assume the light is at the point (0,0,10). What is the *diffuse* intensity at a corner of the square?

c. At what light position is the difference between the intensity at the center of the square and the intensity at the corner equal to 0.01? This could be considered a reasonable distance for the distant light assumption to be valid.

d. Assume that the light is at (0,0,5). What point has the highest specular contribution for the viewer position given above?

e. Now assume a viewer in the direction through the point (5,0,5) but infinitely distant. What changes about the appearance of the specularity: does its center move, does its size change, neither, both?

f. Now assume an infinitely distant viewer, as before, and an infinitely distant light along the z axis. Is there any specularity? If not, why not? If so, where is it?