CPSC 314, Written Homework 4

Out: Wed Mar 31
Due: Mon Apr 12, 5pm
Value: 4% of final grade
Total Points: 90

For any answer involving calculation, although it's fine to show your work in analytical form, the final answer should be expressed as a number to two decimal places.

Any late homeworks put into the box should have the date/time when it was deposited clearly written across the top.

Clipping (32 pts)

1. (16 pts) Clip the line segment with endpoints A = (-1,2) and B = (2,0) to the box (-1,-1), (1,-1), (1,1), (-1,1). Use the Cohen-Sutherland algorithm.

   a) Show the full configuration after clipping against the top of the box with a sketch, including any new points you may need to create. If you need to create new points, give them names in alphabetical order (D, E, F, ...). Compute the exact coordinates of any new points that result from intersecting lines. (Provide the final values as decimal numbers.) Show your intermediate work, including outcodes for vertices.

   b) Same as above, after clipping against the bottom edge.

   c) Same as above, after clipping against the right edge.

   d) Same as above, after clipping against the left and final edge of the box.

2. (16 pts) Clip the polygon with points A = (0, 3), B = (2, 3), C = (0, -2) against the box (-1,-1), (1,-1), (1,1), (-1,1). Use the Sutherland-Hodgeman algorithm to clip the polygon.

   a) Give the vertex list after clipping against the top edge, and draw a sketch showing the configuration including the location of any new points you may need to create. If you need to create new points, give them names in alphabetical order (D, E, F, ...). Use the same labels for the new points in your sketch and in the vertex list. Show your intermediate work, in terms of inside/outside checks for each vertex pair. You do **not** need to compute exact values of the the intersection points.

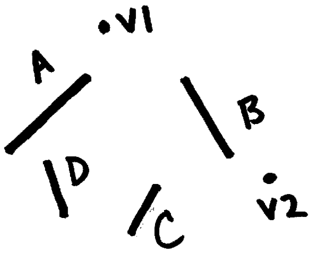
   b) Same as above, after clipping against the bottom edge.

   c) Same as above, after clipping against the right edge.

   d) Same as above, after clipping against the left and final edge of the box.

Visibility (42 pts)

3. (24 pts) Build a BSP tree for the following scene using the polygons (shown as line segments). The cutting plane induced by a polygon should just extend along the line itself. The labelled side of the polygon should be the right child in the tree, and the unlabelled side should be the left child.



   a) Give the BSP tree with the single root node of polygon A, and sketch the entire scene with the addition of the new cutting plane.

   b) Same as above, after adding polygon B.

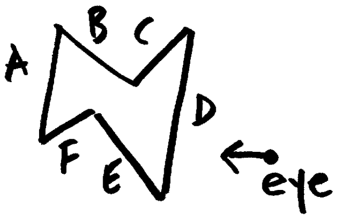
   c) Same as above, after adding polygon C.

d) Same as above, after adding polygon D.

e) Traverse your BSP tree to produce a painter's algorithm ordering from eye point V1. Show your work at each step in the traversal, starting from the root of the BSP tree.

f) Same as above, instead using eye point V2.

4. (2 pts) For the following 2D scene, an eye point is given with respect to an object formed by line segments. Which faces would be removed for the given eyepoint if backface culling were used?



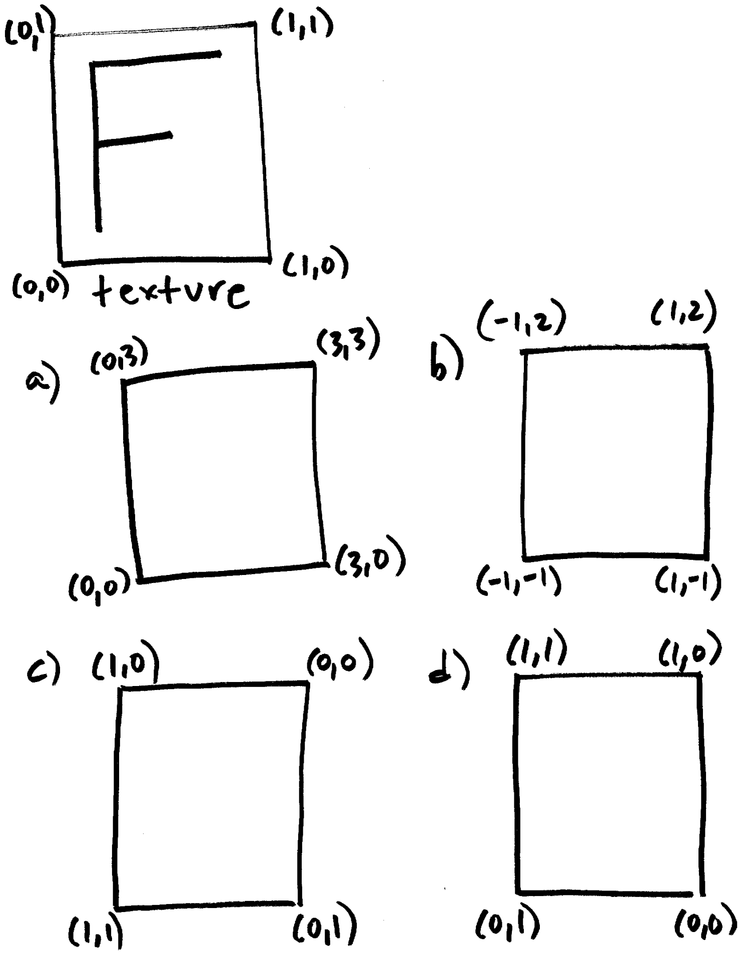
5. (16 pts) You have bought a very cheap graphics card, which has a Z buffer of only 3 bits. You can thus only determine the visibility relationships of objects in your scene at a very coarse resolution: there are only $2^3 = 8$ bins available. These bins are represented as the base-10 integers 0 through 7. You should assume that the general OpenGL perspective matrix was used for projection, with the near plane set to .01 and the far plane set to 10.

a) Give the z-values of the planes forming the boundaries of these bins in DCS, the display coordinate system, which ranges from 0.0 at the near plane to 1.0 at the far plane. That is, what is the value of the plane between bin 0 and bin 1, between bin 1 and bin 2, and so on.

b) Give the z-values of the planes in the camera coordinate system.

Textures (16 pts)

6. (16 pts) In the following figure, sketch the texture (top) as it would appear in each of the rectangles with the specified texture coordinates.


(0,1)
(1,1)
(0,0)
texture
(1,0)
a)
(0,3)
(3,3)
(0,0)
(3,0)
b)
(-1,2)
(1,2)
(-1,-1)
(1,-1)
c)
(1,0)
(0,0)
(1,1)
(0,1)
d)
(1,1)
(1,0)
(0,1)
(0,0)