CS480/CS680 Problem Set 2

Due in class September 29

1. (15 points)
   Give pseudo-code for an algorithm to determine if the vertices of a polygon are given in clockwise (CW) or counterclockwise (CCW) order. Assume a simple front-facing polygon; The polygon may be concave or convex.

   To give an example of acceptable pseudo-code, here is example pseudo-code for the algorithm that determines if a 2D polygon is concave or convex:

```
// Algorithm to determine if a polygon is concave or convex
// Polygon vertices could be provided in CW or CCW order
// Three sequential vertices may be collinear

Input: v1, …, vN              // N polygon vertices
Output: True or False         // True if convex, False if concave

Vector e1, e2;
float z;
int sign_of_sine_theta=0;

// compute cross-product between successive edges
// if sign of all the z values are all the same, then convex
// loop around polygon, taking cross product at each vertex

for (j=1; j<=N; ++j){
   if(j==N)
         k=1;
   else
         k=j+1;
   if(j==1)
         i=N;
   else
         i=j-1;

   e1= vj-vi;
   e2= vk-vj;
   z = (e1.x * e2.y) - (e2.x * e1.y);  // z of cross-product

   if(z < 0.0){
   if(sign_of_sine_theta > 0)
         return False;           // sines with different signs
   else
         sign_of_sine_theta = -1;
   }
   else if (z > 0.0){
   if(sign_of_sine_theta < 0)
         return False;           // sines with different signs
   else
         sign_of_sine_theta = 1;
}
return True;
```

2. (15 points)
Consider Bresenham's algorithm for drawing line segments, where the line color is represented in the range `[0.0,1.0]`. As discussed in lecture, the number of pixels that are turned on per unit length of the line varies, depending on the slope (diagonal lines appear dimmer). We can adjust for this by varying the intensity of the color used in setting the pixels, depending on the line slope. What is the formula we can use to compute the color value, depending on the slope, that would make lines of all slopes appear equally bright?

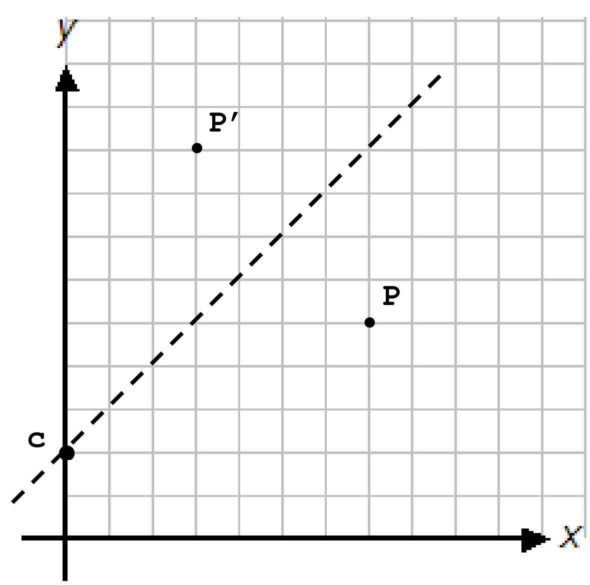
   Assume $0.0 \leq m \leq 1.0$.
   Assume that the diagonal line at $m = 1.0$ is drawn at maximum intensity (1.0).

3. Use quaternions in your answers to the following:

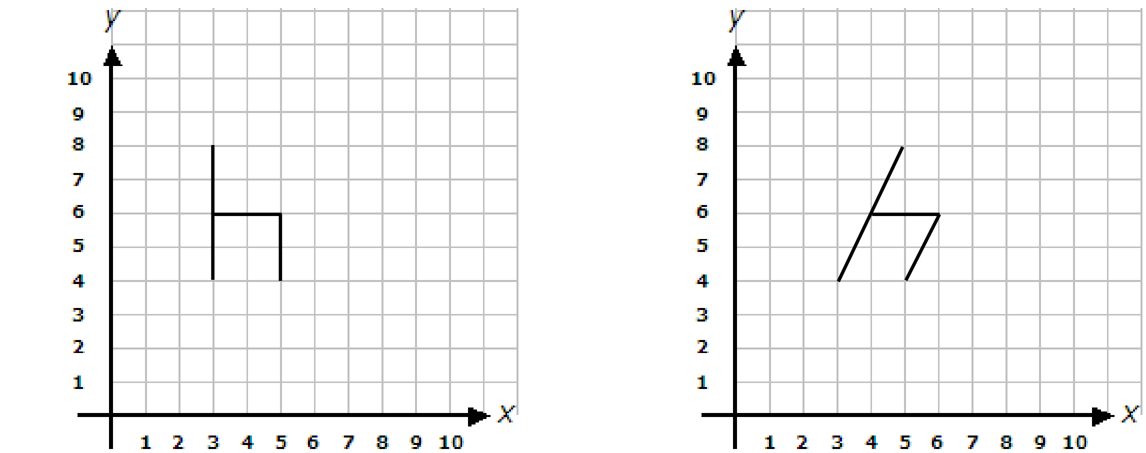
   (a) (15 points)
   Prove that in general two 3D rotations about different rotation axes do not commute. You can assume that both rotation axes pass through the origin.

   (b) (15 points)
   Derive the general conditions under which 3D rotations commute.

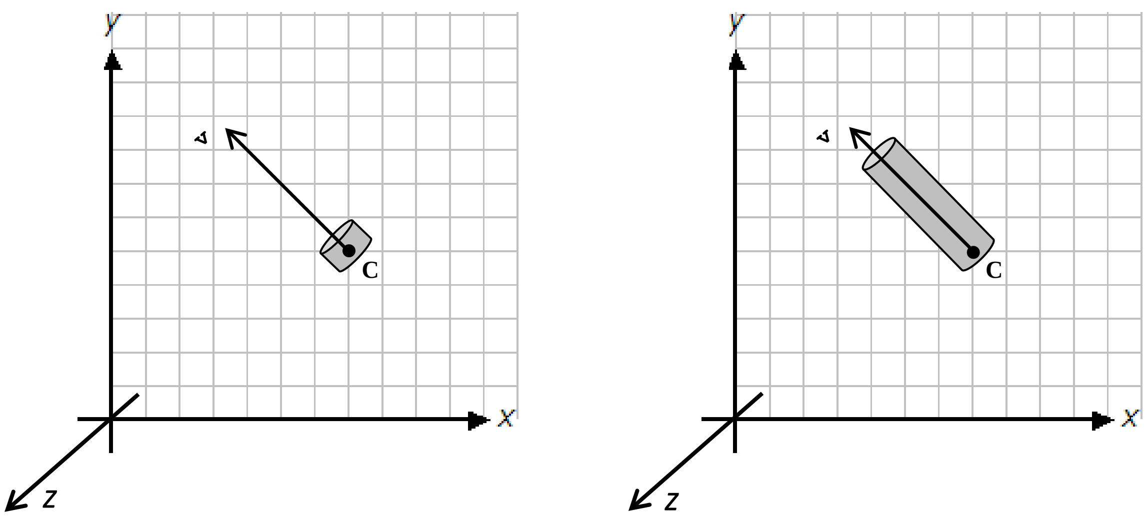
4. (15 points)
Give the composite transformations required to obtain 2D reflection about the line $y = x \tan \theta + c$. Make sure you show your matrices and the order in which they are applied. An example is shown below: A reflection about the dashed line maps the point **P** to the point **P′**.



5. (10 points)
Derive the shear matrix that would transform the block "h" character on the left to the italic "h" character on the right. Replace variables with values for your final answer.



6. (15 points)
Derive a 3D homogeneous transformation matrix to scale along the $\mathbf{v}$-axis with origin $\mathbf{C}$ by $\mathbf{S_v}$. Your solution should not use trigonometric functions for $\mathbf{R_{in}}$ or $\mathbf{R_{out}}$, instead use the orthonormal basis vectors as discussed in class.



7. (CS680 only; 20 points)
Derive a homogeneous transformation matrix that can be used to reflect 3D points about a plane with equation $\mathbf{ax + by + cz = d}$. Your solution should not include the explicit computation of any Euler rotation matrices $\mathbf{R_x}$, $\mathbf{R_y}$, and $\mathbf{R_z}$.

<u>Submission guidelines:</u>
Assignments should be submitted on hardcopy, neatly written or typed at the beginning of class on the due date. Any late assignments should be submitted and time-stamped by staff in the CS main office.