

## CS 480/680 PSet 2: Transformations & Polygons

**Submission guidelines:** Submit on Gradescope. Please prepare your answers neatly written or typed. If you have prepared hand-written solutions, submit a scan or photograph. Acceptable formats are .pdf (preferred), .jpg or .png.

**Point totals:** CS 480: /80 pts; CS 680: /100 pts. Max 10 pts extra credit.

### Question 1: (15 pts)

Give pseudo-code for an algorithm to determine if the vertices of a polygon are given in clockwise (CW) or counterclockwise (CCW) order. You may assume that the polygon is simple and non-degenerate; it may be concave or convex. As an example of acceptable detail, here is example pseudo-code for an 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: v[1], ..., v[N]    // 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= v[j]-v[i];
    e2= v[k]-v[j];
    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;
```

- a) Main question: Consider the case when the input to the algorithm is a sequence of polygon vertices and a point known to be within the interior of the polygon (inside the polygon, and not on the boundary). To simplify things, you may assume we have a function “segsIntersect(p1, q1, p2, q2)” which will return a boolean telling you whether the line segments from p1 to q1 and p2 to q2 intersect.
- b) **Extra credit (5 pts):** Suppose the point in the interior of the polygon is not provided. Describe a strategy for finding such a point. Pseudocode not required.

**Question 2:** (20 pts)

- (a) Write a  $4 \times 4$  homogeneous transform matrix M that when applied to a point  $p = (x, y, z, 1)$  yields  $p' = (x', y', z', 1)$  where

$$x' = x - \sqrt{3}z + a$$

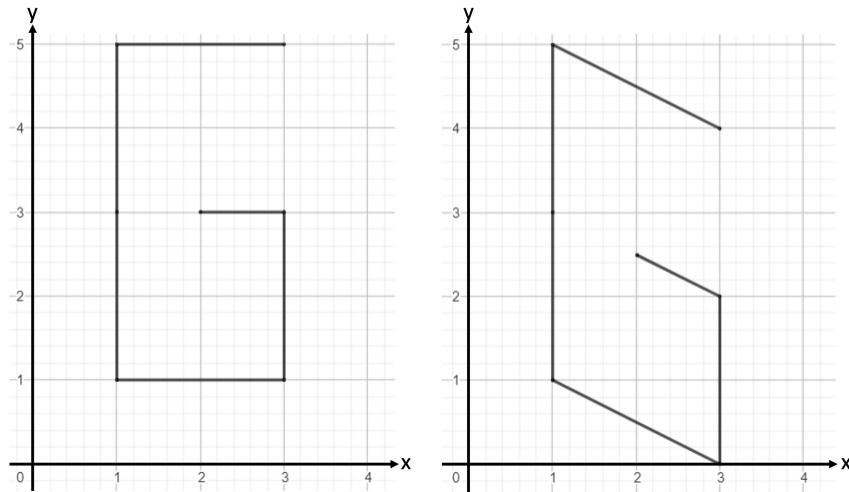
$$y' = -y + b + 2$$

$$z' = \frac{\sqrt{3}}{2}x + \frac{1}{2}z + c$$

- (b) What three or four basic computer graphics transforms occur when applying M to a 3D point? In other words, how can you decompose M into transforms such as scaling, rotation, translation? Give a homogeneous transform matrix for each, and show the order in which they are multiplied.

**Question 3:** (10 pts)

Derive the 2D transformation matrix that would transform the block “G” character on the left to the slanted “G” character on the right. Show your answer first with variables, then replace the variables with values for your final answer.



**Question 4:** (20 pts)

Consider the following unit quaternions

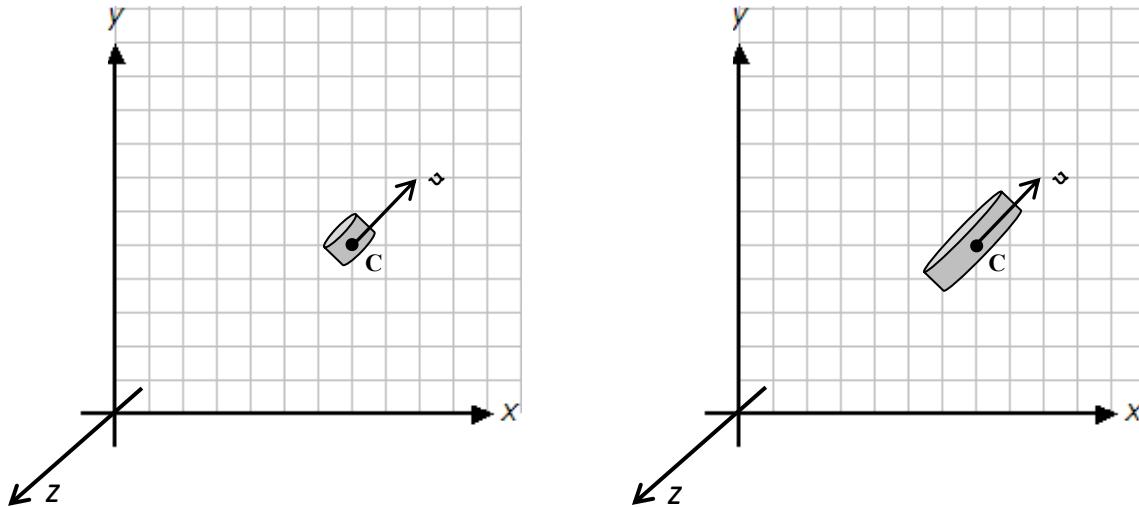
$$q_1 = \left( -\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}} \right)$$

$$q_2 = (0, 0, -1, 0)$$

- $q_1$  represents a rotation of angle  $\theta_1$  about a unit vector  $u_1$ . What is this angle and vector? (Note: there are several acceptable answers).
- $q_2$  represents a rotation of angle  $\theta_2$  about a unit vector  $u_2$ . What is this angle and vector? (Note: again, several acceptable answers).
- Does rotation by  $q_1$  and  $q_2$  commute? Why?
- Extra credit (5 pts):** When do two rotations commute? When do they not? Prove these statements. (Hint: consider the equation for quaternion multiplication).

**Question 5:** (15 pts)

Derive a 3D homogeneous transformation matrix to scale along the  $\mathbf{u}$ -axis with origin  $\mathbf{C}$  by  $\mathbf{s}_u$ . 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.



**Question 6:** (480: 10 pts, extra credit; 680: 20 pts, required)

Derive a homogeneous transformation matrix that can be used to reflect 3D points about a plane with equation  $\mathbf{ax} + \mathbf{by} + \mathbf{cz} = \mathbf{d}$ . Your solution should not include the explicit computation of any Euler rotation matrices  $\mathbf{R}_x$ ,  $\mathbf{R}_y$ , and  $\mathbf{R}_z$ .