

Computer Graphics Lab

Lab 4 Manual

[A separate source file is provided for your understanding about transformation. Have a look and understand the code.

The only difference is `glutSpecialFunc(key)` call where a callback function `key(int key, int x, int y)` invoked.

Add source codes of Task 1,2 & 3 and hand in.

Mention your reg.no. in at least one filename]

*** You may take the help of the textbook but not any online resources*

*** Use black background and white for scan converting color*

*** Use the coordinates limits large enough to execute enough transformations*

Suppose, you're given a quadrilateral in counter-clockwise direction,

$$V = \{(4, 8.5), (1, 7), (4, 10), (7, 7)\}$$

This looks like an arrowhead (), isn't it nice 😊?

*[Remember: the trigonometric function(s) take input or give output in radian, **not** in degree. So, think about the conversion]*

Task 1: **30%**

Let's say, our up-key(↑) means counter-clockwise rotation and down-key(↓) means clockwise-rotation. For each press, we can assume, the angle changes by 1 degree.

Your task is to implement this rotation (both CW & CCW) with respect to the point (4,17).

Task 2: **30%**

You can move the triangle up, down, left, or right by pressing the arrow key (just translation).

Look, initially, the arrowhead quad is facing up.

After you press each arrow key once, the arrowhead will start moving one unit facing that direction.

Perform this transformation.

[Hint: Think about mirroring]

Task 3:

30+10%

Let's say, the center of the arrowhead is (4,8.5). We will move in the same manner as Task 2. But we will define some scaling factors here.

Let's say, initially the distance between the origin and the initial arrowhead center is d . After we move our arrowhead to a new position, let's say our arrowhead center comes to a point (x',y') . The new distance between the origin and the center becomes d' . We define our scaling factor as:

$$S_x = S_y = \frac{d'}{d}$$

After, moving to a new location, the arrowhead will change according to the scaling factor it achieves for the new position.

Implement this scaling transformation.

The arrowhead will be in white while $|S_x, S_y| \leq 1$ and red elsewhere. Can you implement this one also?