

# Computer Graphics

## Assignment 1

-Archit Sangal (IMT2019012)

### Answer to Questions:

1. I don't think that there is much of a difference between the 2 approaches. Though there can be some differences with respect to the efficiency of execution.

In approach 1, we can calculate the transformation matrices for each primitive. After placing the shape in its final location, we can use that space for some other primitive and store the vertices, as they will not be changed.

In approach 2, we need to store the centroid and matrices (or parameters that help in making the matrices) as they can be used in calculations further, and hence, for each primitive, we need a transformation matrix stored.

Approach 1 is kind of a part of approach 2 itself.

2. We use mouse down event listener (DOM EventListener for HTML Javascript - refer to reference 5.) for implementation of "picking" using mouse button click button, which is given as follows-  
`canvas.addEventListener('mousedown', function(event){ onmousedown(event);});`
3. If *m* is pressed, calculate  $(++m\%4)$ , and do all the necessary initial stuff. The use of mode 0 and mode 3 is not that important. We can directly start from mode 1 and after mode 2, we can go to mode 1. If we just want to minimize click events, we can click once and then use mouse drag like events like in google draw.
4. The use of centroid is not that critical in translation, but for rotation and scaling, it is pretty essential. For rotation, we require a point of rotation, and for scaling a point of scaling around which it can scale itself relatively. As a user, we are more accustomed to using the point of importance as the centroid owing to the software products that we use like PowerPoint, Google Docs, etc. We could use other points of rotation and follow the algorithm, they will work fine too.

## References:

1. <https://stackoverflow.com/questions/9031041/calculate-minimum-bounding-rectangle-of-2d-shape-by-coordinates>
2. <https://gamedev.stackexchange.com/questions/116419/how-do-i-calculate-a-vertexs-position-on-the-cpu>
3. <https://github.com/Amit-Tomar/T2-21-CS-606> (Example 3 and 5 for base code)
4. A lot of guidance from Amit Sir.
5. [https://www.w3schools.com/js/js\\_htmlDOM\\_eventlistener.asp](https://www.w3schools.com/js/js_htmlDOM_eventlistener.asp)