

GRAPHICS - ASSIGNMENT II

Sutherland Hodgeman Polygon clipping:

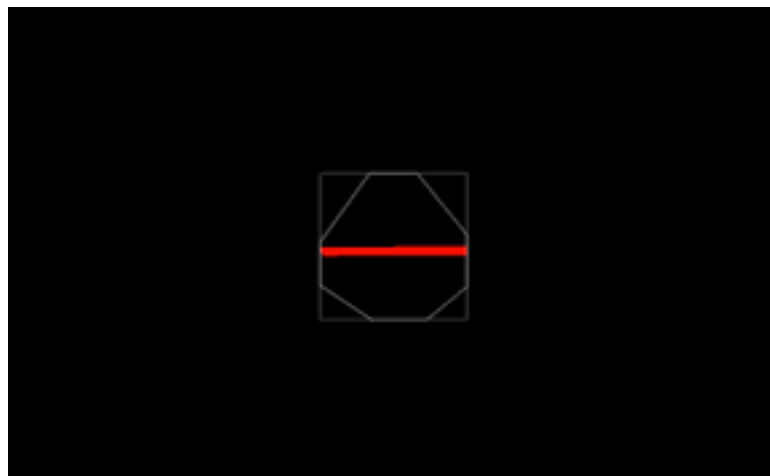
I have clipped the polygon at the 4 boundaries of the window. Left, right, top and bottom respectively. The vertices of the polygon are taken as the input and the computation occurs by checking whether they are inside or outside the polygon. Four cases occurs when applying this algorithm.

Both the vertices may lie outside the boundary which we are considering . They are as follows.

1. Both the points are outside the polygon. Eliminate both the points.
2. Both vertex are inside the polygon. save the incoming vertex.
3. In \rightarrow out ... save the intersection point.
4. Out \rightarrow In ... save the intersection point and the In point.

Filling the polygon: (Boundry-fill)

For filling the polygon boundary fill algorithm has been used. I does the filling using recursion. I checks whether the point meets the color of the boundary or the colored pixel. If it doesn't meet the condition it colors the pixel by the specified color.

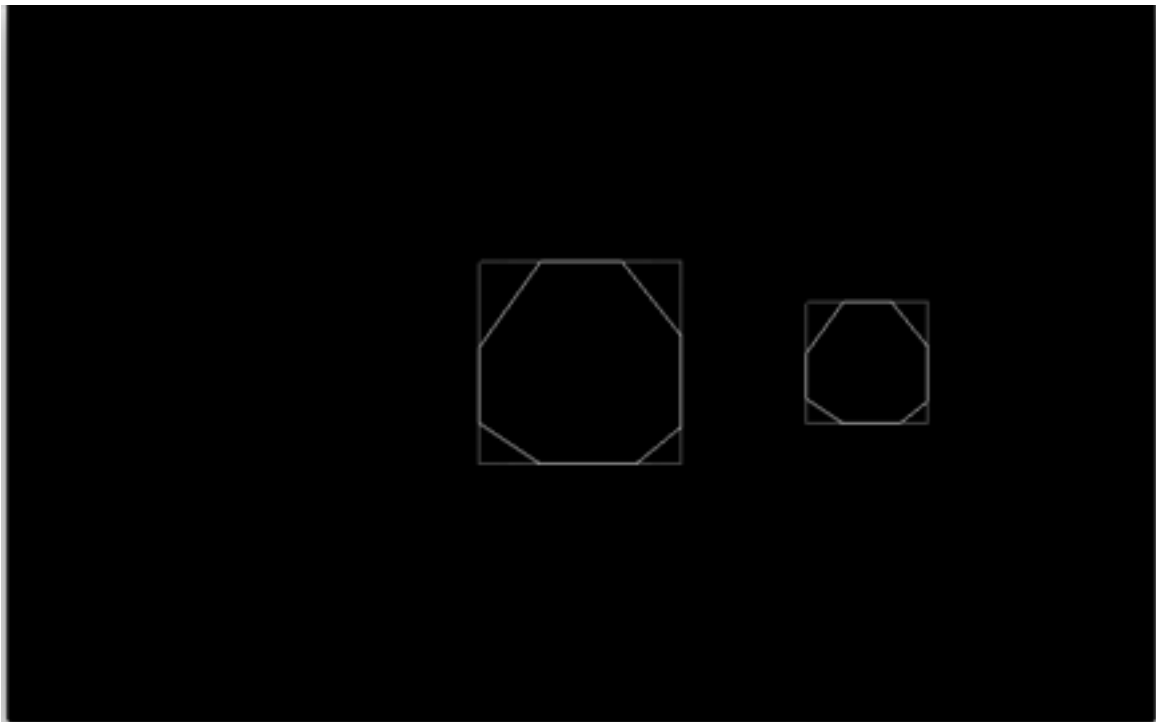


Window-to-Viewport:

For this transformation, initially shift the object to origin and then scale the object accordingly. Then translate it back to the viewport co-ordinates which are specified. We can select this option in the menu which has been created.

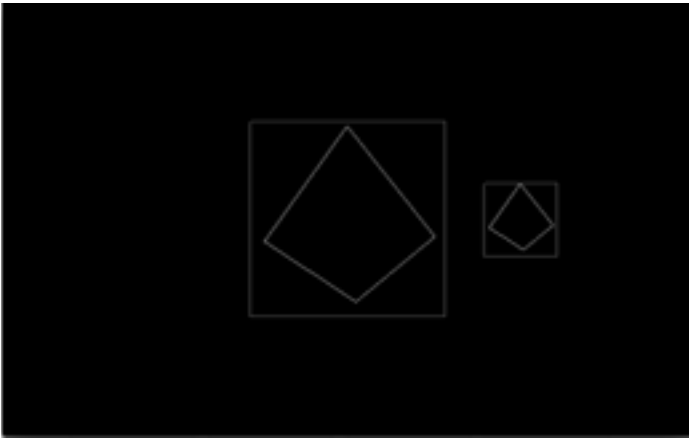
This is the pseudo code for the scaling and transformation.

```
scale_x = (vp_x_diff*1.0)/ x_diff;  
scale_y = (vp_y_diff*1.0)/ y_diff;  
for(int i=0; i<left_click_counts-1; i++) {  
    vp_x[i] = vp_xmin+((x_points[i]-xmin)*scale_x);  
    vp_y[i] = vp_ymin+((y_points[i]-ymin)*scale_y);  
}
```



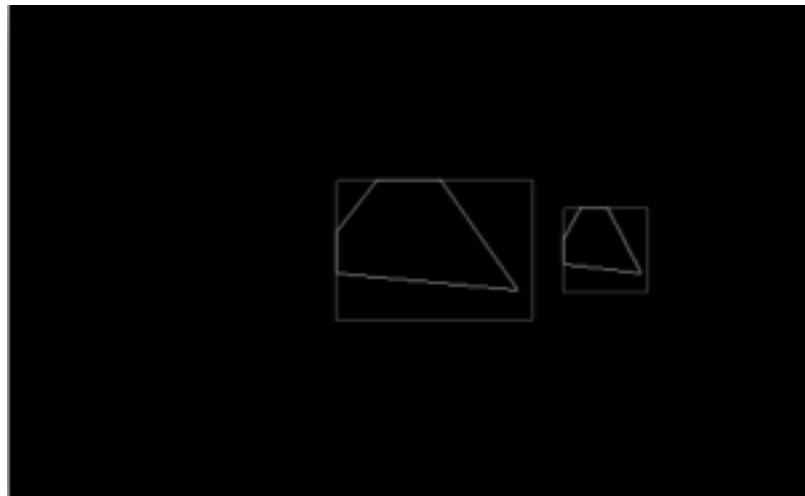
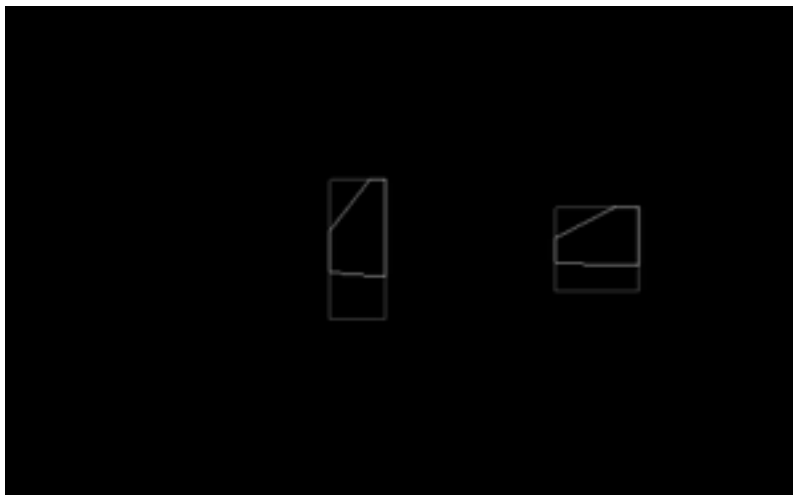
Zooming:

Zooming has been done by changing the window size, clipping the polygon again and transforming the new clipped polygon to the viewport. This can be viewed by selecting 'window-to-viewport along with zooming' option from the menu. Zooming can be seen by pressing 'z' and 'x' buttons on the keyboard. Both zoom-in and zoom-out options have been implemented. Both the images are attached.



Strength Right:

This effect can be achieved by stretching the windows towards right and left, clipping the polygon accordingly and transforming it to viewport. They have provided user option to stretch right and left using the keys 'c' and 'v'.



Panning:

In my demo, the window can be moved up, down, left and right and we can view the corresponding changes in the viewport. The polygon is clipped based on the window movement dynamically and represented in the viewport. I am attaching the images for pan up, left in the report. We have use keys 'a', 'w', 's', 'd' to see this feature in the app by selecting appropriate option from the menu.

