CS460 HW2 Report

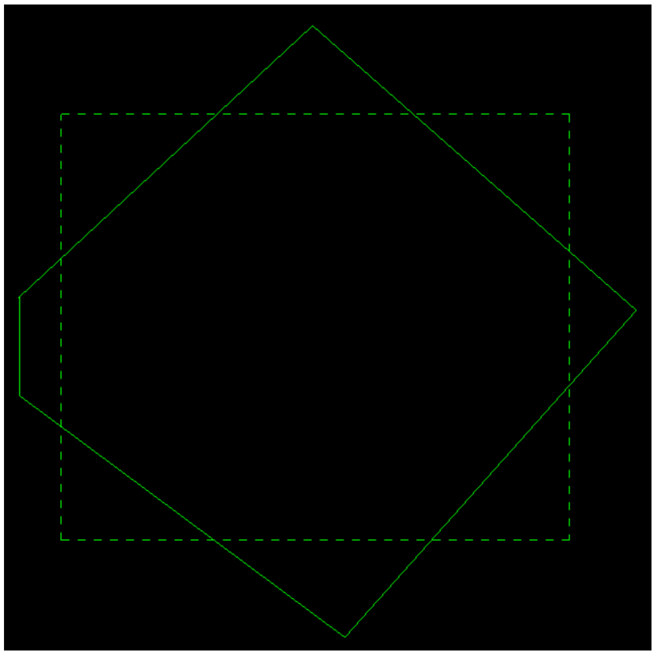
Umut Cem Kayaalti / B00694107

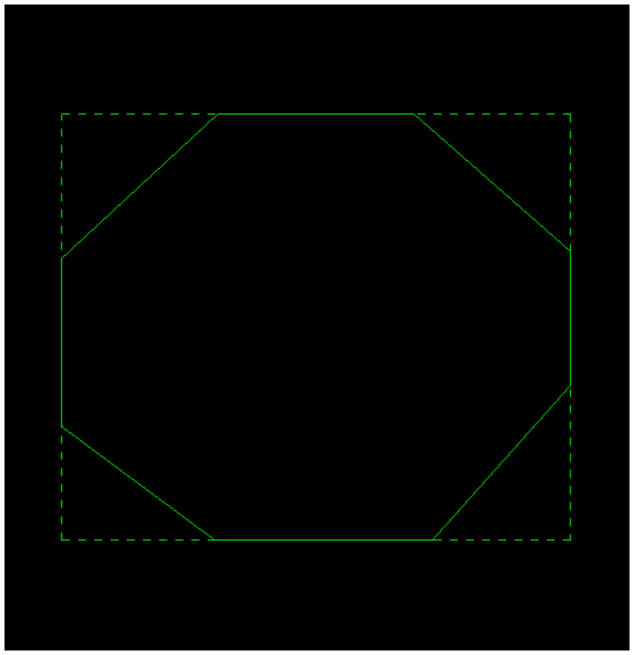
In the homework for mouse drawing I used the code I have written in the previous homework. I first draw windows for both clipping and viewport. I modified the mouse functions for the drawing. Now after right clicking to stop the drawing if you right click again you will be able to access a menu. This menu has three options one for clipping, region filling and window to viewport mapping. If you click left mouse click after stopping drawing it will reset everything.

Clipping

In the homework for clipping I applied the Sutherlands algorithm. To do that I had to check for the clipping windows borders in the order of left, right bottom, top. First any line that is outside the left border got clipped to their left border intersection. If both end of the lines are out of the left then those vertexes get eliminated. If going from inside the left border to outside then we create a new vertex at the intersection of the border. If the line is coming from the outside border o to the inside then we create a vertex on the border and also use the vertex on the inside. Then use these vertexes on the right, bottom and top using the same procedure. This results in our clipped polygon. Its worth noting that given shape is assumed to be a polygon and treated as such so even if the shape given is not closed loop it will close it by assuming.

Clipping Examples

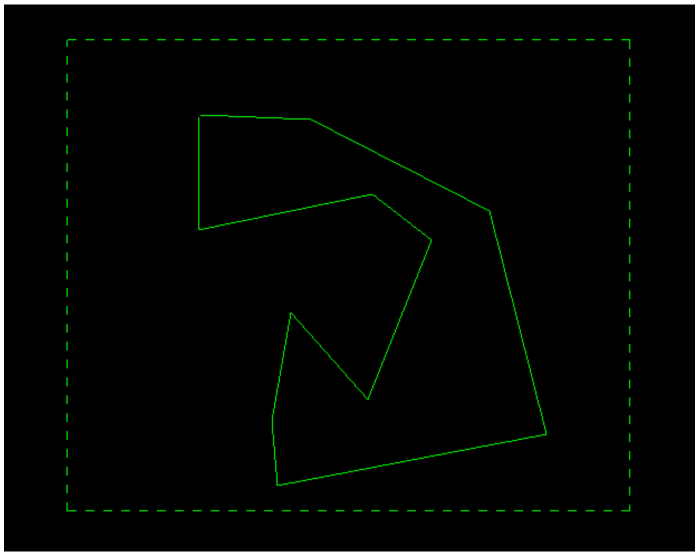
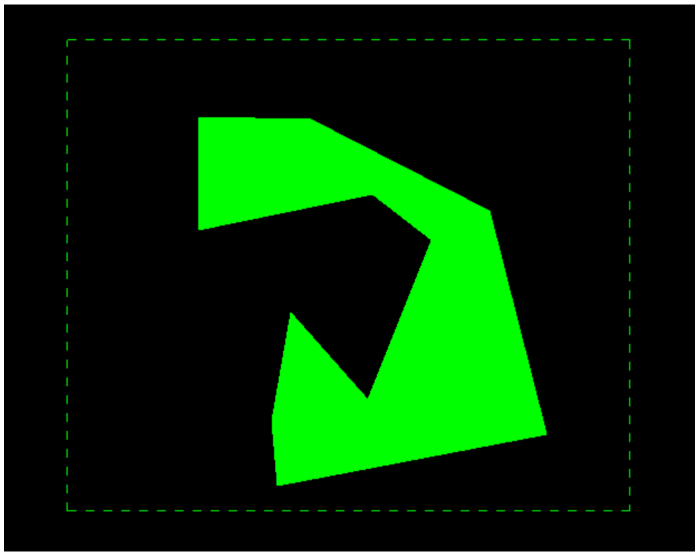
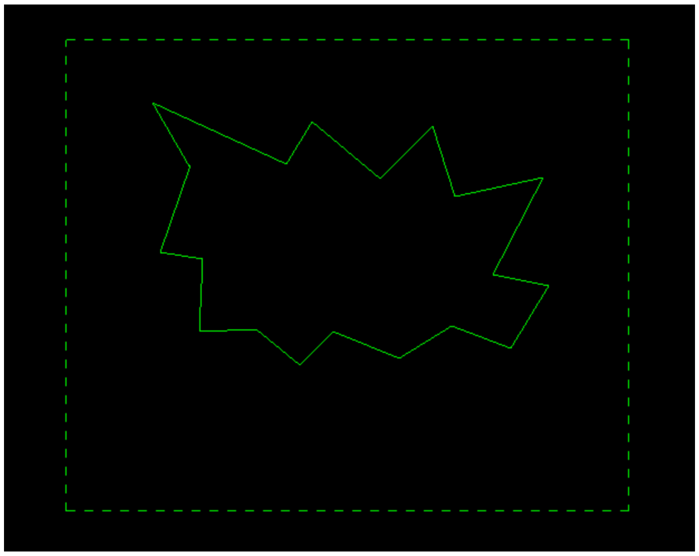
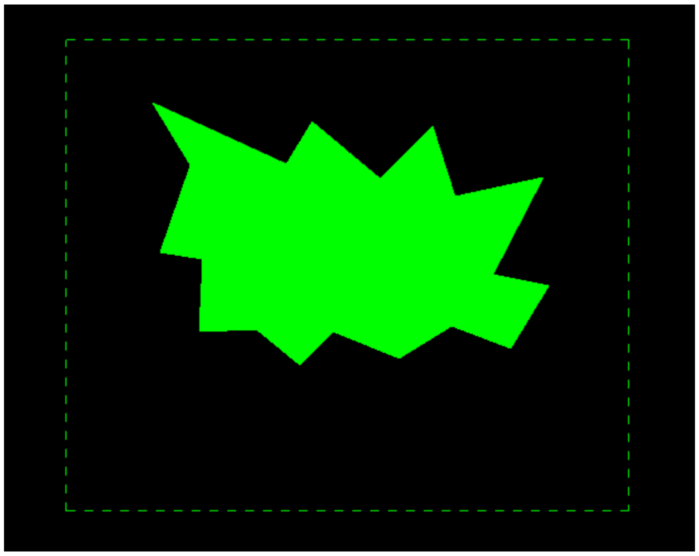




Region Filling

For region filling part of the work I used Scan-Line polygon filling algorithm. To do that I had to start from the bottom of the clipping window and increase the x by 0.001 to fully paint the polygon. Then for each horizontal line I had to check with which lines inside the clipping window does our line intersect with. The intersection points were added to a vector and create a line among every two intersect. I had to make sure that the intersection was happening between the lines biggest and smallest x values as otherwise the intersection is not inside our polygon. There was also problem with triangular shapes top or bottom were getting counted in wrong numbers as there were need to add some corners twice or once. To fix that I only added the intersection if it is bigger or equal to the smallest y value of that fraction of line. If intersection happens at the biggest y value I skipped adding. This stopped the adding of the intersection more than once as one lines smallest y value was occasionally neighboring lines biggest value and got added again. And it happens that there is no need to draw a line for the tip of the triangle so not adding it was working.

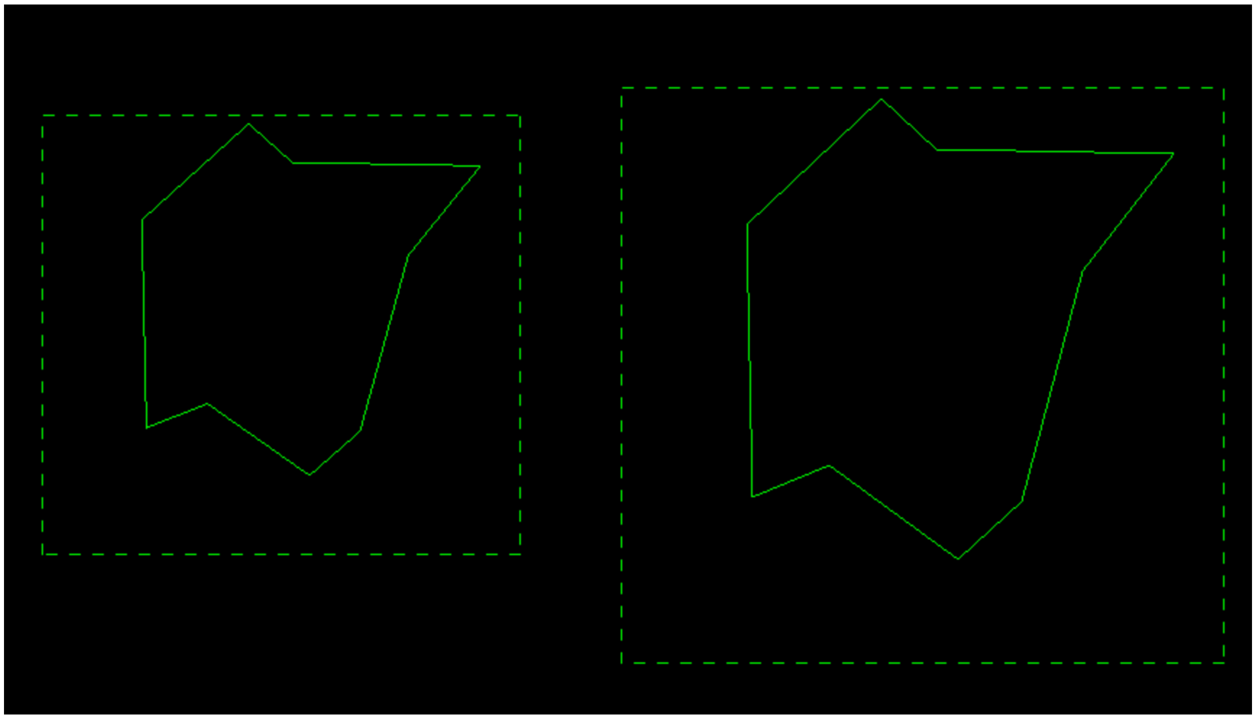
Clipping Examples

Window to Viewport Mapping

When it came to mapping the window to viewport I had to calculate how far away from the left side of the clipping window the current given line was. After that I had to adjust the distance to the viewport by dividing the distance to the width of the clipping window and multiplying it with the width of the clipping. This gave its starting leftmost x coordinate and did the same for the y as well with height. After that I had to adjust the length of the line by again dividing by the difference between the x values of the line by width of the clipper window and multiplying with the viewport width. Done that same for y with heights instead of width. Finally using that adjusted differenes and starting our new starting points I just had to create new lines and draw them.

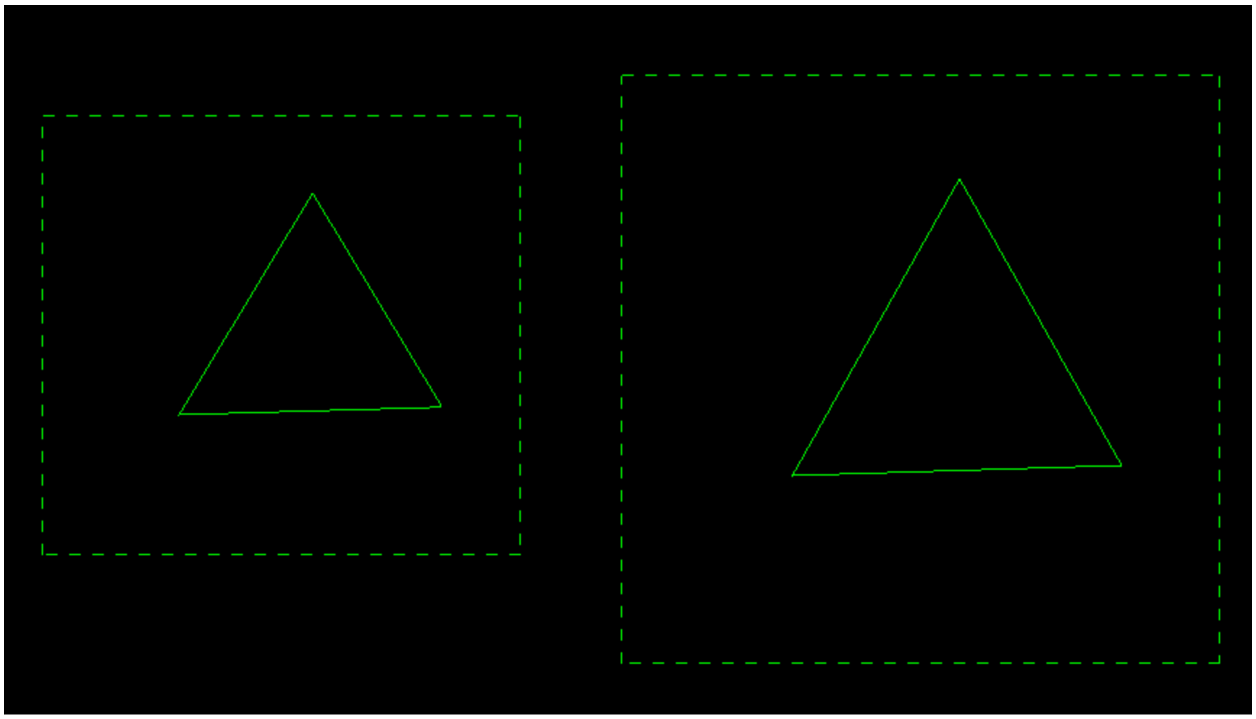
Window to Viewport Mapping Examples

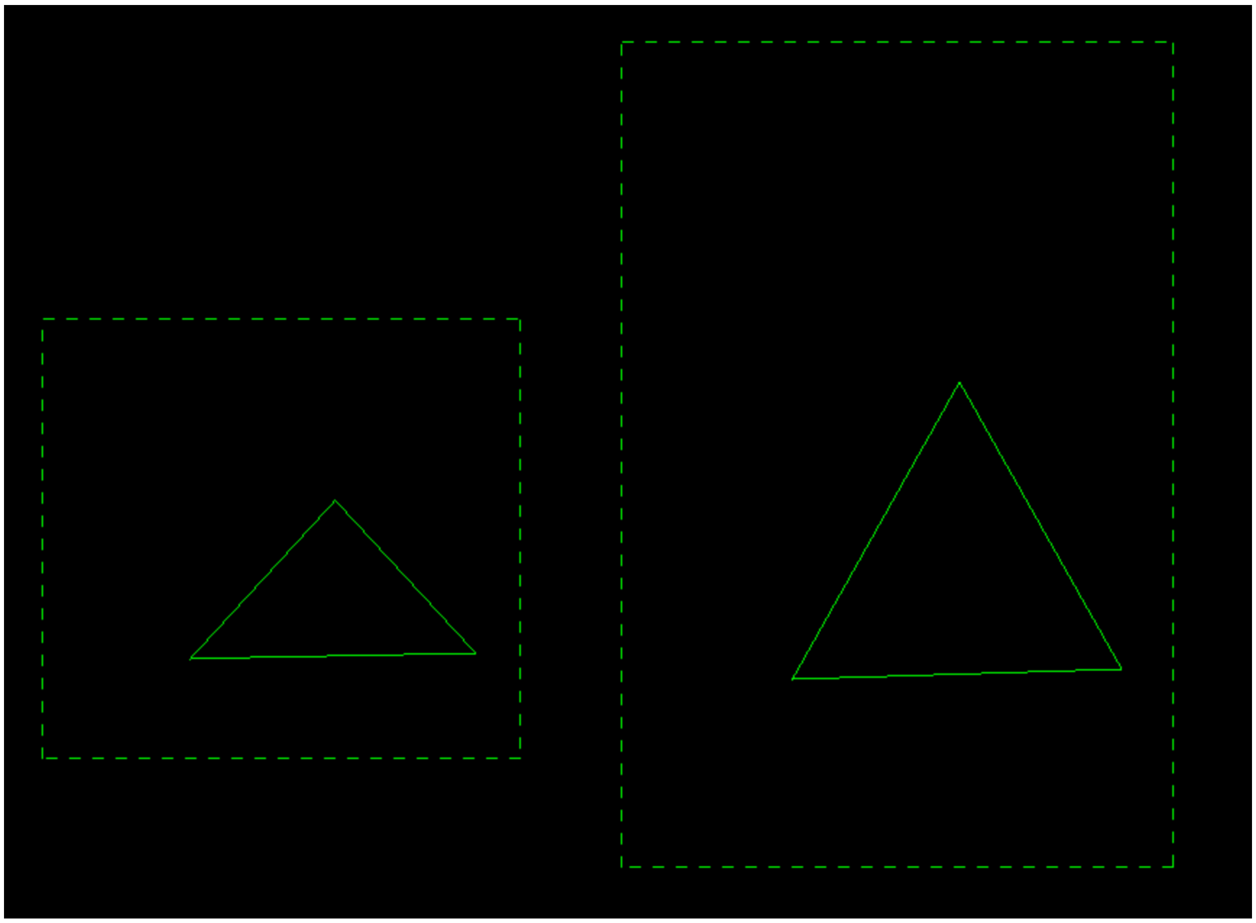


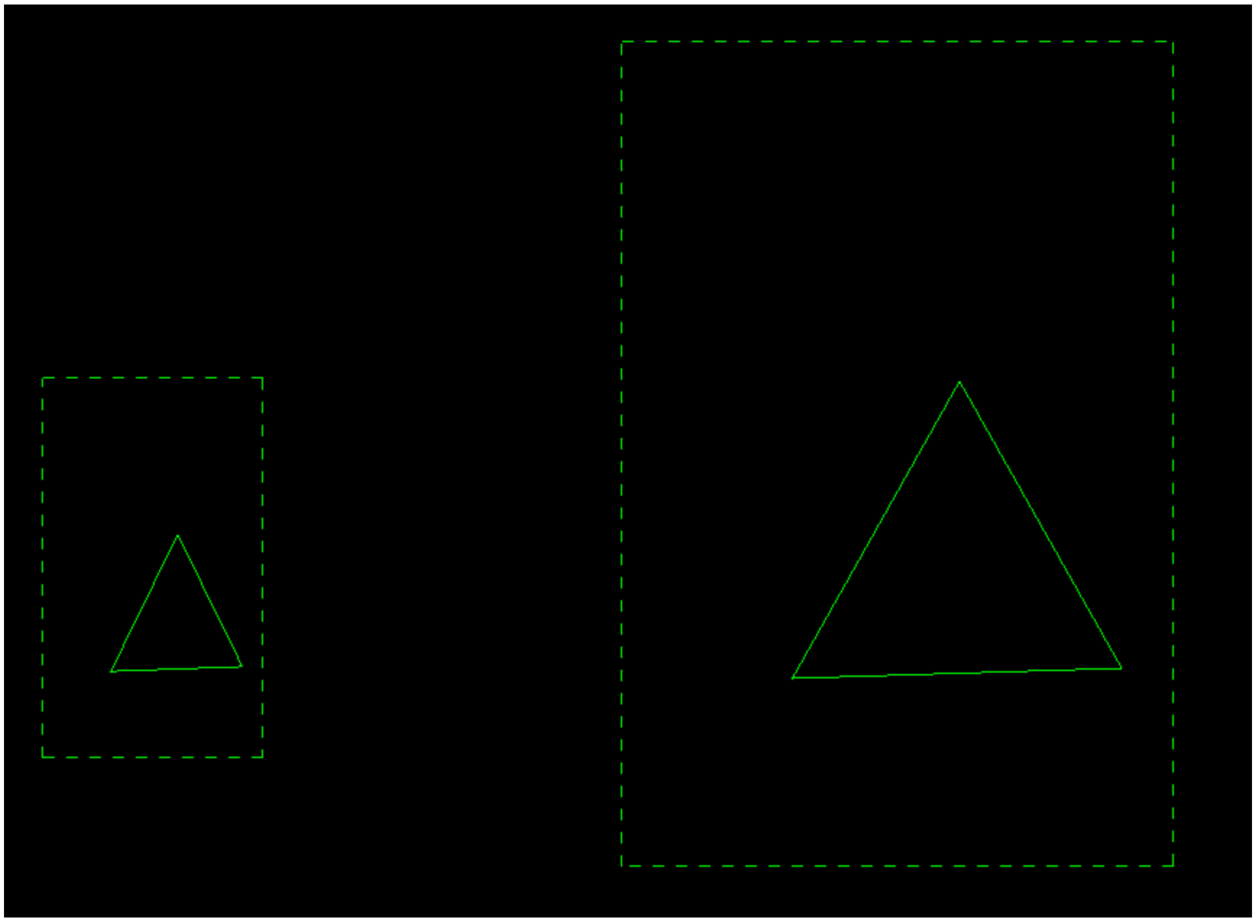
Scaling The Clipping and The Viewport Windows

For scaling the windows I inserted a new state for mouse clicking. When left mouse button clicks on the corners of one of the windows it enters in a scaling state. As long as the left mouse is still pushed down it will call a function to update the shape of the targeted window in to its new shape by using the new top right coordinates. After updating the window it will call the window viewport mapper to update the shape inside the viewport as well.

Scaling The Clipping and The Viewport Windows Examples







Moving The Clipping and Viewport Windows

For moving the windows I inserted a new state for mouse clicking. When clicked on the top middle of any one of those windows it will track the movements of the mouse as long as the left button is still clicked. The new mouse coordinates will be subtracted from the previous top middle coordinates of the clicked window and difference will be added to each corner thus moving the windows.

Moving The Clipping and Viewport Windows Examples