 Computer Graphics –

Line Drawing Assignment

# Objective

The objective will be to implement a fully generalized line drawing algorithm. You will need to implement the parametric line drawing algorithm and write code that can draw any line between two points on the raster surface. Just like assignment one, this will be done in a **WIN32 CONSOLE APPLICATION** that includes the **RasterSurface** files. You may want to use the same project from assignment one as a starter project for this since you likely wrote several helper functions that can also be used here. If you are going to reuse that project, make sure to make a copy of the project so that you have the original code from completing assignment one in case you need it.

The end goal of this assignment is to have a function that can take in two (X, Y) points and draw them as well as all of the points between them to form a visible line on the raster display. You will also write code to test this line drawing function to ensure it can draw any combination of lines. Your function signature may end up looking similar to this:

* void LineFunctionName(unsigned int \_x1, unsigned int \_y1, unsigned int \_x2, unsigned int \_y2);

You may create a structure that stores the two points together, or anything else that may help you to organize your code.

To test this function, you will draw an octagon by connecting points clockwise around the shape and then connect the points of the octagon with diagonal lines through the middle of the shape.

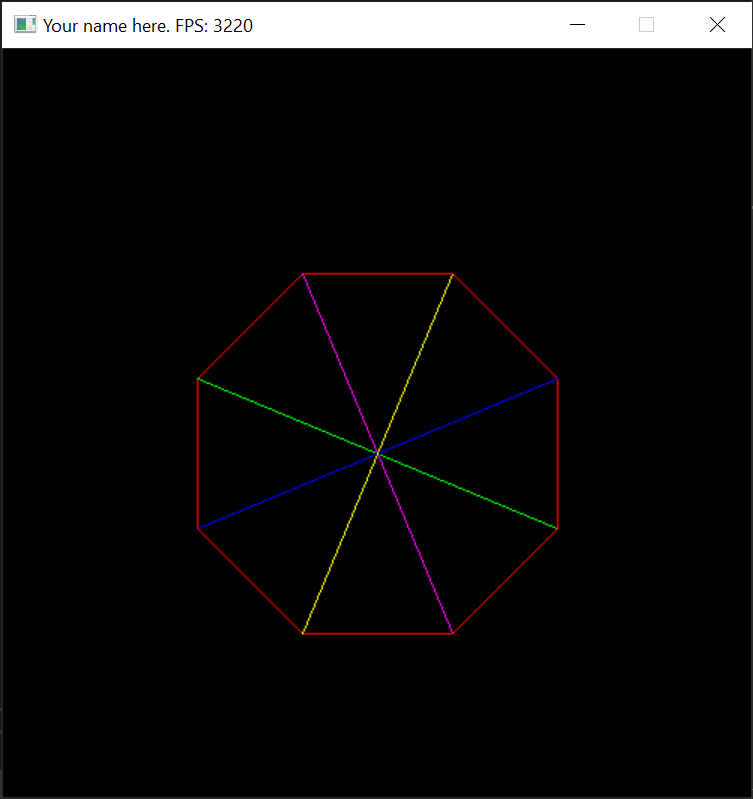
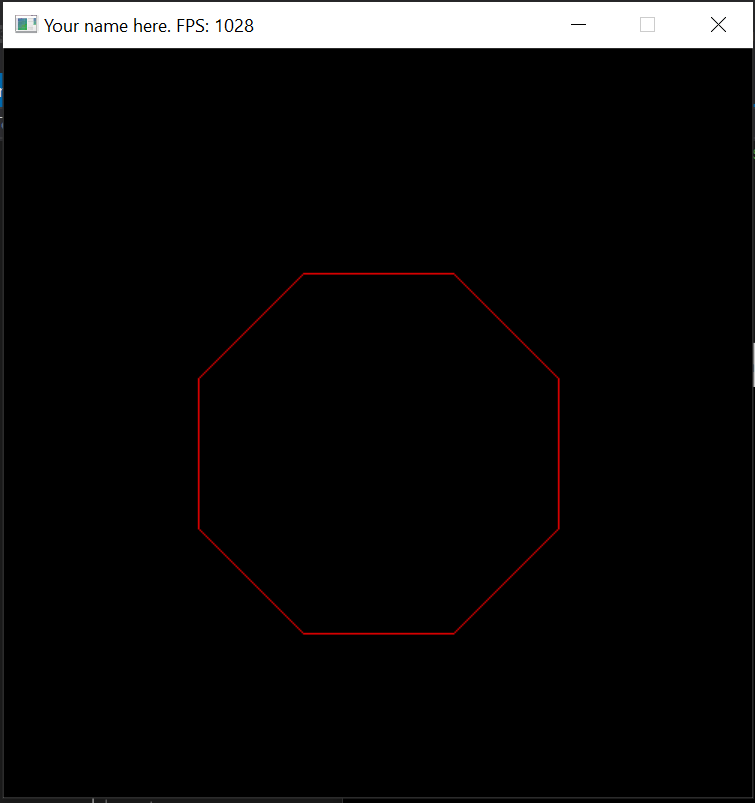
# Grading Breakdown

* 8.3% per line that is successfully drawn. There will be a total of 12 lines.
* If the implemented line drawing algorithm doesn’t utilize linear interpolation, then the maximum possible score is a 50%.

# Guidance

1. To get full credit for this assignment you will need to implement the parametric line drawing algorithm, which means at least one use of linear interpolation should be present in your line drawing code.
2. You will need code for drawing an individual pixel just like in the first assignment. A drawing loop will also be needed for when you are ready to test your line drawing function by drawing the octagon.
3. Start by just implementing code to match the pseudo-code from the slides. Remember that the pseudo-code is written expecting a line that travels from left to right and may also be expecting a specific up or down direction as well.
4. Once you have the pseudo-code working, you can start testing out other lines of the octagon that don’t travel left to right or have other challenges associated with them and begin debugging and fixing your line drawing implementation to handle those cases.
5. To keep length measurements simple, if you have the horizontal and vertical sides as 100 pixels long then the diagonal lines should be about 70 pixels long to keep the octagon equal sized on all sides.
6. Once you have the octagon fully drawn, the lines crossing through the middle should just use start and end points that you have already plotted out as part of the octagon.

# Example Images



# Submission

Programming assignments will be turned into FSO. Follow the directions listed there carefully, failure to do so can significantly impact your grade in a negative way. If you did not entirely write your own code for this assignment, don’t turn it in. MOSS will be used to check the similarity of submitted code.