

## Rasterization

### Line rasterization

For the line rasterization I used the given pseudocode from the class, transcribed, and modified it.

For lines that start from a higher y point and are supposed to be drawn from top to bottom, I reverse the process. I start the process from the y with the lower value and decrease x instead of increasing it, making the line be drawn backwards.

When the slope of the line is greater than 1, I used an alternate case of Bresenham's Algorithm, where I flip between where x's are used with y's and vice versa. This process allows for the lines to be drawn more cleanly.

### Circle rasterization

Similarly, to line rasterization, I used the given pseudocode code, and modified it.

The only real modification I had to do was for CirclePoints which mirrors where pixels are placed. Where I used the given points of x-naught and y-naught and used them to translate the points in the function to their proper location.

After using and modifying the given pseudo-code this resulted in the final image:

