COSC 4370 Homework 1 Report

Khoa Huynh 1514353

1 Introduction

The assignment requires the rasterization of an ellipse described by the equation (x/12)^2 + (x/6)^2 = 64^2 where y >= 0. The bmp header was provided along with the setPixel function.

2 Method

I modified the main function of the program to draw the first quadrant of the eclipse. I added local variables to the function to main in order to keep track of the variable. I also made the width and height of the canvas variables to easily modify them if needed. I also made compiled the color values into a struct for ease of use.

The general idea behind rasterizing the eclipse is the same as the reading provided. I used the equation (ax)^2 + (bx)^2 – (ab)^2 = 0 to determine if a midpoint between 2 pixels is inside or outside of the ellipse. After we have determined if the midpoint is inside or outside of the ellipse, we pick the pixel nearest to the arc and move on to the next midpoint. Rasterizing and ellipse is different from rendering a circle because we cannot take advantage of the 8 way symmetry due to the different lengths of the axes of the ellipse. We must manually rasterize the bottom part of the arc where we have to select between the South East pixel and the South pixel.

3 Implementation

Because the assignment only requires us to rasterize the first quadrant of the ellipse we can frame the origin at the bottom left of the canvas and adjust the width and length of the canvas to fit the arc. I calculated the values of the a and b from the given formula and made them into variables along with the variable a^2 and b^2 for ease of use.

3.1 Dividing into 2 regions

As discussed above, ellipses do not have 8 ways symmetry like circles, so we need to write code for rasterizing the whole first quadrant of the ellipse. We must now pick the point where the curve starts to intersect the South Eastern pixel and the Southern pixel. I choose to follow the example of the reading and pick the point where the derivative of the arc is equal to negative one. To calculate this, we compare the b^2*x to a^2*y. If the b^2*x is less than the other equation, then the slope is still greater than negative one and where are still selecting between South East and East. Otherwise, we are now rasterizing region 2.

3.2 Rasterizing region 1

For region 1, we implement the same algorithm as the provided reading. The initial condition of the algorithm is to check if the x = 0 and y = a is inside or outside of the circle. If it is inside we select the South East pixel and if it is outside we select the East pixel. After that we select the next mid-point until the slope of the arc is greater than or equal to -1.

3.3 Rasterizing region 2

The initial condition of the region 2 is to check to see if the next mid-point is inside or outside of the circle. Because the arc is now having a more pronounce curve down towards the y axis we must now select between the South East pixel and the South pixel

4 Result

The output of the program is a bmp file that renders the ellipse. Some artifacts can be seen at the border between the 2 regions, and I think that the algorithm to rasterize the second region can be improved if I have to time.