COSC 4370 – Homework 1

Name: Caitlin Dooley PSID: 1389102

September 2021

1 Problem

This assignment requires the rasterization of the upper half of an ellipse. Given the ellipse $(x/12)^2 + (y/6)^2 = 64^2$, the algorithm must rasterize the section where y>=0. Relevant sections of the textbook "Computer Graphics Principles and Practice - Foley et. al" referenced for this assignment are section 3.2 “Scan Converting Lines” and section 3.3 “Scan Converting Circles”.

2 Method

The only file necessary to modify was main.cpp and no libraries or header files could be added. The main.cpp template included a loop that iterated across the image drawing a white line at a constant y. Using this template as the base for the solution, the method was to draw each half of the semi-ellipse starting from the highest point (farthest point on the ellipse from the x axis) and set the pixels for each side out from there utilizing the provided loop with some modifications.

3 Implementation

First, the size of the image needed to be modified to fit the result. Making the image 2000px wide and 1000px tall allowed for some padding around the resulting semi-ellipse. Using the ellipse equation provided, the width and height of the semi-ellipse would be 1536px and 384px respectively. Solving the ellipse equation for y provided the y axis calculation to set each pixel based on x as the loop iterates horizontally across the image. When the constant y is replaced in the bmpNew.set_pixel() function inside the loop, instead of resulting in a straight line, the righthand half of the desired semi-ellipse was output. In order to draw the left section of the semi-ellipse, another set_pixel() function was added inside the loop, but rather than iterating from 0 and up for the x axis value, another iterator was added to move in the opposite direction.

For clarity, the first set_pixel() function resulting curve was set to be colored red and the second set_pixel() function’s resulting curve was colored green. In order to line the curves up to connect and create the full ellipse half, the red curve was shifted over 768px which is half of the width of the ellipse. Because the resulting curve was a bit jagged looking with gaps at each end where the curve was the steepest, the i and j

iterators were changed from int to float and each incremented by 0.5 instead of 1.0 to smooth the curve. The loop was also modified to finish at 768 rather than run the full width of the image to remove the trailing straight lines around the curves. Finally, the resulting ellipse was translated horizontally and vertically to land in the middle of the image rather than lining up with the lower left edge.

Because the edges of the ellipse were still not clean and solid, another loop iterating up the y-axis and solving for y was necessary. With one loop drawing each half of the semi-ellipse based on the x value, and another loop drawing over these halves based on the y value, the resulting ellipse has even, solid lines from end to end.

4 Results

The result of the program was a .bmp file, which when viewed through an image viewer, shows the image with the half ellipse shown in green and red on a black background.