**PA3: Polynomial Area Approximation Algorithm**

Developer: Eric Gelphman

University of California, San Diego

Version: 0.5

This simple, yet handy C program calculates the area of any polygon in the plane. The polygon can be convex, concave, regular, irregular, and have any number of sides, so long as its sides do not intersect and the figure is closed. The program uses a variation of Green’s Theorem to do the computation. uses Green’s Theorem. I chose to write this program using C instead of Java because, quite simply, Java is not well optimized for algorithmic/functional type programs like this. This is mostly because C’s syntax is less clunky, and C/C++ programs run much faster than Java programs that do the same thing because there is less of a semantic gap between computer hardware and software. Moreover, most computers (including Macs), come with C built-in, meaning there is no need to download a development kit (JDK).

To do its computations, the program needs the (x,y) coordinates of all of the vertices of the polygon. The program prompts the user to enter them (x coordinate 1, then y coordinate 1, then x coordinate 2, and so on) through the console. The coordinates can be integers or decimals. The program then does its computation, then outputs the area of the polygon through the console (Note: when C was first invented in 1972, console input/output did not yet exist). This program is best compiled and run via command line (Terminal for Mac, Windows 10 machines also have a command window. You can download and install a Unix terminal on a Windows PC fairly easily). You may need to download the gcc compiler for C (just Google it and then download it). If you are unable to use command line, Eclipse also works for C and C++ programs, and several other C IDEs exist.

This program is currently a prototype under development for version 1.0. Version 1.0 will approximate the area of any closed curved shape in the plane using the same algorithm. But, instead of entering the (x,y) coordinates for the verticies, the user will instead enter the equation(s) for the boundary of the shape. The program will then read and identify the function, and take an (x,y) point at a fixed step distance along the curve and add it to the algorithm. This should quickly produce an approximation with acceptable precision to shapes whose area cannot be found using double integrals easily(or at all).

For Those Unfamiliar With Green’s Theorem:

<http://mathworld.wolfram.com/GreensTheorem.html>

The images on the next page show the derivation of the algorithm.

