## CSE328 Fundamentals of Computer Graphics: Programming Assignment ONE

## Programming Assignment ONE (10% of the entire course grade, total 500 points are used here for the grading scheme of this programming project. Note that, the due time is 2:10 pm on Wednesday, February 25, 2015)!

**Please note that, all programming assignments/projects in this course will be graded based on (1) the correctness of program functionality, (2) the efficiency of program execution, and (3) the quality of programming codes (i.e., whether they are understandable with proper comments)!**

**The main objective of this programming assignment is to learn basic and key components of two dimensional graphics techniques and apply them (i.e., line drawing, scan-conversion, two-dimensional transformations, etc.) to solve real-world problems in practice. Please write an OpenGL program that implements the drawing of a polygon boundaries as well as its interior and be able to handle standard two-dimensional transformations and manipulations. In particular, the entire project must consist of the following six major functionalities (you should use OpenGL utility library to handle issues of user interfaces and interactions, but you MUST use draw-point function for drawing tasks):**

**PART-1: (100 points) Draw all the line-segments of any user-specified polygon using the mid-point algorithm (detailed in our class lecture and in the required course textbook). Your program should allow users to interactively enter positions (including both x coordinate and y coordinate) of a set of vertices that define the polygon via mouse buttons. Your program should try to use different buttons for different purposes. For example, left mouse click indicates a new vertex, and right mouse click indicates the end of input process, hence a closed polygon should appear when you click on your right button in this scenario, please feel free to make your own choices;**

**PART-2: (100 points) Detect if the input polygon is a simple polygon by checking whether any two line-segments intersect with each other besides their end-points. Please provide users any meaningful visual feedbacks whenever the self-intersection occurs. Your program should have an option that discards any non-simple polygons and only keeps the simple polygons for the following tasks (in PARTS (3), (4), (5), and (6) explained below);**

**PART-3: (100 points) Scan-convert the simple polygon and fill the interior of the simple polygon;**

**PART-4: (50 points) Deform the polygon by interactively selecting vertices using mouse buttons and moving them to new locations on your computer monitor. Please note that, your program should always repeat PARTS (1), (2) and (3) whenever you deform the polygon, i.e., redraw the new line-segments, detect and discard the non-simple polygon, and re-fill the interior, etc. Once again, for example, you can use left mouse clicks to choose vertices and right mouse clicks to move the selected vertex to the current location as indicated by current mouse position, please feel free to provide visual feedbacks to users in order to enhance the interactivity;**

**PART-5: (100 points, 20 points each for translation, rotation, scaling, shear, and reflection with respect to y = -x line) Define your local 2D coordinate system first and then perform two-dimensional transformation operations (translation, rotation, scaling, shear, reflection with respect to y = -x line) on the specified polygon with respect to your newly-defined 2D coordinate system based on any user-specified quantities (such as translation vector and/or rotation angle). In principle, you can use either mouse buttons or command-line input to provide these quantities. For example, you can use mouse operations to interactively translate the polygon to its current position, and the same idea can be generalized for rotation, scaling, shear, and reflection operations. For example, the distance between the current position of your mouse click and the user-specified origin can be used as the scaling factors that control the enlarging and/or shrinking effects along either x or y axis, respectively;**

**PART-6: (50 points) Polygon morphing is to transform the starting polygon P1 into the ending polygon P2. Your program should be able to handle the simple task of 2D polygon morphing which will be explained in the lecture. First, your program should allow users to interactively specify two polygons (both P1 and P2) that have the same number of vertices. Then, your program should permit users to interactively establish the correspondence between two sets of polygonal vertices. Again, you should provide users meaningful visual feedbacks while you are specifying the individual correspondence. Finally, your program should be able to animate the morphing process using the simple linear interpolation technique that is explained in the lecture. By running your program, users will be able to see that the first polygon is morph-ed to the second polygon.**

**Please note that, this assignment deadline is at the start of our lecture at 2:20pm on Wednesday, February 25, 2015!!!**

**The project submission is done electronically. Please note that, CSE328 TA will announce the e-submission procedure in class during the next three weeks!!! After submitting your programming assignments/projects via e-submission, you must also write and submit a hard-copy one-page (or two-page) document explaining in details how we can execute you program and verify the above functionalities properly!**