**Problem Statement**

The goal of this assignment is to create an OpenGL program that lets the user select three random points using their mouse. The program uses these points to draw a triangle and three circles: an inscribed circle, an excircle, and a nine-point circle.

**Proposed Solution**

In the design phase of the assignment we discussed the requirements that the program needed to meet. These requirements helped us determine how we were going to implement the solution. After the user selects the three points, we need to draw the triangle and calculate three different circles. The algorithms for each of the circles are different from one another. In the sections below we will outline each of these algorithms and the steps we took to complete our program.

**Strategies and Algorithms**

The requirements of the assignment helped structure the strategies that we used for our solution. We started by setting up the initial framework for the program. We used the Canvas class for this purpose. Canvas gave us quite a bit of functionality through the tools that it provides. All we needed to do was instantiate a Canvas object to utilize them. This made setting up the framework much easier than building everything from scratch.

After the framework was setup, we focused on the three random points that the user selects. As the user performs a left mouse click, the x coordinate and y coordinate (corrected to viewport position) are saved. This information is stored in a struct that holds three Point2 data types from the canvas class.

Now that the coordinates of the three points are stored in the struct, we can draw our triangle and begin the calculations for the three circles. The triangle is the easiest shape to draw. We achieve this by using the moveTo and lineTo methods provided by the canvas class. Now it is time to calculate the three circles.

The assignment describes three types of circles that need to be drawn: an inscribed circle, an excircle, and a nine-point circle. Each circle is unique and requires separate calculations. The defining characteristics of the circles are described below:

* The inscribed circle is a unique circle that’s defined by the three points tangent to the three sides of the triangle.
* The excircle is a unique circle where the circumference of the circle passes through the three points selected by the user.
* The nine-point circle as the name describes is defined by nine distinct points. The nine points are as follows: the midpoints of the three sides, the point of intersection from each vertex to the midpoints, and the midpoints of the lines joining the orthocenter to the vortex.

**Mathematical Techniques**

The algorithm for the inscribed circle requires the location of the three points tangent to the sides of the triangle. Let’s call them R, S, and T. Next we will call the three points selected by the user A, B, and C. The point R is defined as A + La. The point S is defined as B + Lb. The point T is defined as T = A - La. Now that we have the three points, the center of the circle is calculated using our calcCenter function. The last thing we need is the radius of the circle. This is simply calculated by passing in the three points to our radCirc function. The circle is then drawn using the drawCircle function.

The algorithm for the excircle was the easiest circle to draw. The first thing we do is calculate the center of the circle by passing in the three user defined points to our calcCenter function. Next we calculate the radius by passing in the three points to our radCirc function. The circle is then drawn using the drawCircle function.

The algorithm for the nine-point circle is the most complex of the three. However, not all nine points are required to draw this circle. We only need three points. To achieve this we calculate the three midpoints. This is done by adding the points together and dividing by 2. After this, we calculate the center of the circle by passing in the three points to our calcCenter function. Next we calculate the radius by passing in the three points to our radCirc function. The circle is then drawn using the drawCircle function.

The calcCenter function finds the center using the following equation:

S = A + ½(a + (b\*c/\*c)). The radCirc function finds the center of an excircle using the following equation: Radius = |a|/2

**Program Output/Results**

Our program uses the algorithms we just described and displays a triangle, an inscribed circle, an excircle, and a nine-point circle. Each circle is color coded to identify a specific circle. The green circle is the inscribed circle, the yellow circle is the excircle, and the cyan circle is the nine-point circle. This lets us know that our algorithm worked to the assignments specifications.

**Summary**

This assignment asked us to implement a program that calculates a series of circles based on three input points from the user. As a team, we discussed the requirements that the program needed to have. Once the requirements were outlined, we began developing the code required for each circle. Different mathematical calculations are used to determine where each circle needs to be drawn. Once the calculations are complete, the circles are displayed on the screen so the accuracy of the results can be verified. Finally, our program is an efficient solution that fully meets the requirements specified by the assignment.