MAT 2170: Laboratory 9

Key Concepts

1. Random Number Generator
2. Writing & Using methods

Instructions

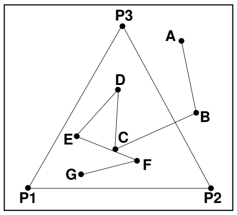
- Create these projects in a `lab9` directory.
- Use methods and named constants as appropriate.

Exercise

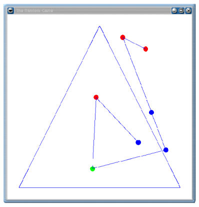
1. (*Chaos Game*). The Chaos Game begins with a triangle determined by three vertices, `P1`, `P2`, and `P3`, within a graphics window. Traditionally, a random point `A` is chosen to start, however, we will place `A` in the window's center. The next point is then generated by randomly selecting one of the triangle vertices and moving halfway toward it. We make that point the new current position, `B`. We generate point `C` by randomly selecting a triangle vertex and moving halfway toward it from `B`, and so on until we have generated as many points as the user wished. For example, in the figure below on the left, the vertices were selected in the order: P2, P1, P3, P1, P2, P1. So `B` is half the distance from `A` to `P2`, `C` is half the distance from `B` to `P1`, ..., and `G` is half the distance from `F` to `P1`.

   In the sample windows below right, the vertices were chosen in the orders given above them:

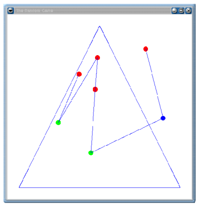
   P2, P1, P3, P1, P2, P1

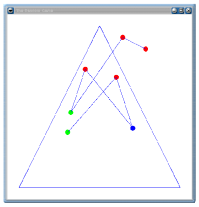
   3, 2, 2, 1, 3, 2

   2, 1, 3, 3, 1, 3

   3, 1, 3, 2, 3, 1

   Have the plotted circles change color depending upon which vertex they move toward: if they were caused by moving toward vertex 1–green; vertex 2–blue; and vertex 3–red.

   You will need two pieces of information from the user: the margin size (how close the triangle vertices should be to the window edges), and how many points the user wants plotted. Extend the DualSliderProgram with slider ranges set to:

   ```
   setRangeA(0, 8);     // margin, in tens of pixels
   setRangeB(1, 100);   // number of points, in thousands
   ```

   Multiply the margin (slider A) by 10, and the number of points (slider B) by 1,000.

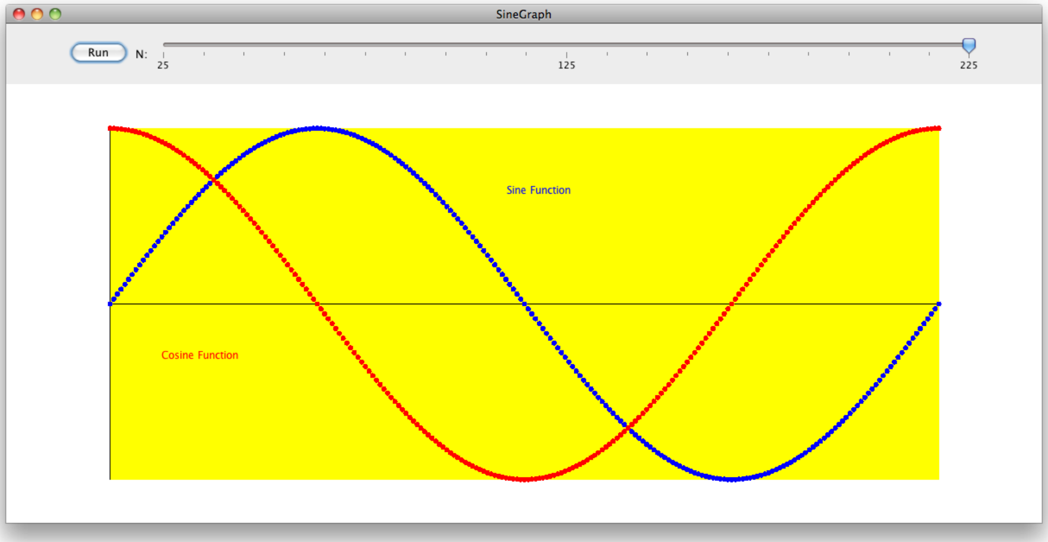
   Create and use at least two methods, `midPoint()` which calculates the midpoint between a point and a chosen vertex, and `addFilledCircle()` similar to `CreateFilledCircle()` but adds the `GOval` to the window without returning it. A general outline of your program:

   - Create a random number generator capable of randomly choosing one vertex of the triangle at a time
   - Create the vertices for the triangle, and draw it
   - Create the first point in the center of the window with a size of 1 pixel and yellow in color. None of the circles should have a black border.
   - Generate and draw points according to the method described above. Do **not** draw lines between the points.

   Recall that the midpoint between two points (x_1, y_1) and (x_2, y_2) is $(\frac{x_1+x_2}{2.0}, \frac{y_1+y_2}{2.0})$.

2. (Calculating π, Page 214–215, Exercise 3) Calculate an approximation for π using a random number generator. Develop and utilize *at least* two methods for this project:
   (a) The method `throwDart()` should return a random (x, y) coordinate (with each of x and y a floating–point value lying between -1 and 1).
   (b) The method `isInsideCircle()` should evaluate whether the (x, y) point generated above lies inside the unit circle; i.e., whether $x^2 + y^2 < 1$.

3. *Sine and Cosine Graphs.* Write a program to display the plots of the sine and cosine functions in the range $[0..2\pi]$, as illustrated below.



   (a) Use a slider program, with a range from 25 to 225, which provides the number of points to plot.
   (b) Set the graphics window to 1200 × 600.
   (c) Make the background rectangle 80% of the window size.
   (d) Draw lines for the axes: one horizontal for the *x–axis*, and one vertical at the left side of the background for the *y–axis*.
   (e) You are required to write and utilize methods:
      i. `translateToWindowX()`: takes a value in the interval $[0, 2\pi]$ and translates it to fit in the background rectangle
      ii. `translateToWindowY()`: takes a value – from either `Math.sin()` or `Math.cos()` – and translates it to fit in the background rectangle
      iii. `addFilledCircle()`: similar to `CreateFilledCircle()`, but *adds* the `GOval` instead of returning it.
   (f) Make the waves different colors. Add labels (with color matching the plotted points) identifying the graphs.

Finishing Up

1. As usual, publish to your web site, submit an electronic copy of your lab and hand in printouts before lab 10.