CS 30 Lab 4 — Graphics and Random Numbers

This lab introduces graphics objects and random numbers. You are encouraged to work with a partner, and to talk things over with your lab-mates. Don't hesitate to call me or Dan over for help or answers to questions.

---

1. Adding some randomness to your programs can often spice them up by making their behavior less predictable. It's easy to generate and use random numbers with Python built-in `random` library. Let's experiment interactively with some of the available functions. Start Python and then type `import random`. The functions that will be most useful to us are:

   - `random.randrange(`*a*`)` returns a random integer from 0 up to but not including *a*
   - `random.randrange(`*a*`, `*b*`)` returns a random integer from *a* up to but not including *b*
   - `random.uniform(`*a*`, `*b*`)` returns a random floating-point value between *a* and *b*
   - `random.choice(`*list*`)` chooses an element from a list or sequence at random.

   Test these functions with various inputs until you feel comfortable with the way they work.

2. The graphics library that comes with your textbook makes it easy to draw images. To load in the library, type

   ```
   from graphics import *
   ```

   This form of the import command makes it unnecessary to prefix the name of each library function with "`graphics.`" like we did above with the functions from the random library. Next, let's create a drawing window:

   ```
   win = GraphWin("My drawing", 400, 300)
   ```

   This creates a Python object (officially of type `graphics.GraphWin`) which represents our drawing canvas, and stores it in the variable `win` so that we can easily refer to it by name. You should see a rectangular window pop up, which will be 400 pixels wide and 300 pixels high. Now you're ready to try out the various graphics functions available.

   Read through the documentation handout that I passed out, as well as sections 5.3 and 5.4 of your textbook (pages 125-132), trying out some of the examples listed.

3. An archery target consists of a central circle of yellow surrounded by concentric rings of red, blue, black, and white. Each ring has the same "width", which is the same as the radius of the yellow circle. Write a program that draws such a target. Hint: Objects drawn later will appear on top of the objects drawn earlier.

4. Write a program that draws some sort of face.

5. Write a program that draws an abstract artistic design, some of whose elements are determined randomly (*i.e.* by using Python's random number functions). This could include randomly choosing the positions of points, lines, or other shapes, or randomly choosing colors.