CSE 231 Spring 2016

Programming Project #2

Assignment Overview

This assignment involves coding and testing of a program using Python control (*if, for, while*) from Chapter 2. In addition you will use some Python libraries to draw randomly colored figures.

This assignment is worth 20 points (2% of course grade), and must be completed before 11:59 PM on Monday, January 25th.

Assignment Deliverable

The deliverable for this assignment is the following file:

`proj02.py` – the source code for your Python program

Be sure to use the specified file name and to submit it for grading via the **handin** system before the project deadline.

Background

You will use `turtle` module (*Appendix B*) to draw rectangles and squares and use the `random` module to select random colors to fill the figures. Turtle is a 2D graphics package that uses a Cartesian coordinate system and a "turtle," which you can imagine has a pen attached to its body. The turtle can move around the plane, drawing as it goes.

Program Specifications

1. Your program will prompt for two numbers:
- Length (an integer).
  (The length will be the starting value for the length of a side of a square or the radius of a circle)
- Selection (an integer)
  (1 selects squares; 2 selects circles)
- Error checking: you will check that the length is greater than 10 and that the selection input is valid, i.e. 1 or 2. If an error is found, print an appropriate error message and end the program without drawing any figures. Do your error checking after *both* values are input. (Assume that the user will input an integer; if something other than an integer is input, it is fine for Python to halt your program with an error—we will learn a good way to handle that later.)

2. Your program will draw a series of figures (squares or circles, depending on the selection) that are decreasing in size and will stop drawing when the size is less than 10. After drawing a figure decrease the size by some value that you have encoded in your program (you get to choose the value).

3. After drawing all the figures, delay the termination of the turtle window with these two lines of code (they assume that you have included the `time` module).

```
time.sleep(5)
turtle.bye()
```

4. As an academic requirement your program must use at least one `if` statement, at least one `for` statement and at least one `while` statement.

Assignment Notes:

Getting Started

1. Divide-and-conquer is an effective problem solving strategy. Consider taking these steps in designing your program. Get each step working before moving on to the next. Start with the `line.py` sample program that we provide (rename it to `proj02.py`)
    a. Modify the program so that it draws one square (instead of the line)—this should have a random color. Hint: use `for`.
    b. Add input to specify the length of the side of the one square.
    c. Wrap a `while` loop around that code so that it will draw a series of rectangles decreasing in size until the length of the side is less than 10. (Ensure that each rectangle has a random color.)
    d. Use copy and paste to copy that code within your program and modify it to draw a series of circles decreasing in size. (The resulting code will draw rectangles and then draw circles.)
    e. Add input to select squares or circles. Then wrap an `if-else` statement around your code to use the input to choose to draw squares or circles.
    f. Add error checking. Hint: wrap an `if-else` statement around your code. Check one of the conditions first, e.g. length is greater than 10. Once that is working add a check for the selection. Consider using the Boolean operators `and`, `or` and `not`.
2. Test your program. Begin using values from the Sample Interaction, but do more tests than that.
3. If your turtle drawing window doesn't disappear, try one or both of these steps:
    a. Enter `turtle.bye()` in the IPython shell window.
    b. In the upper right corner of the IPython shell window is a small icon that yields a drop down selector with choices "Interrupt Kernel" and "Restart Kernel". Try "Interrupt Kernel" and if that doesn't work "Restart Kernel" will.

Sample Interaction

```
In [128]: runfile('/Users/enbody/Documents/cse231/SS16/Projects/Project02/Admin/proj02.py',
wdir='/Users/enbody/Documents/cse231/SS16/Projects/Project02/Admin')

Enter the starting length > 10: 5

To draw squares enter 1; to draw circles enter 2: 1

Error in starting length.

In [129]: runfile('/Users/enbody/Documents/cse231/SS16/Projects/Project02/Admin/proj02.py',
wdir='/Users/enbody/Documents/cse231/SS16/Projects/Project02/Admin')

Enter the starting length > 10: 100

To draw squares enter 1; to draw circles enter 2: 4

Error in figure selection.

In [130]: runfile('/Users/enbody/Documents/cse231/SS16/Projects/Project02/Admin/proj02.py',
wdir='/Users/enbody/Documents/cse231/SS16/Projects/Project02/Admin')

Enter the starting length > 10: 100

To draw squares enter 1; to draw circles enter 2: 1

In [131]: runfile('/Users/enbody/Documents/cse231/SS16/Projects/Project02/Admin/proj02.py',
wdir='/Users/enbody/Documents/cse231/SS16/Projects/Project02/Admin')

Enter the starting length > 10: 100

To draw squares enter 1; to draw circles enter 2: 2
```