CSE 231 Fall 2009

Programming Project #2

Assignment Overview

This project focuses on rendering different colors and some simple graphics using python libraries. It is worth 20 points (2% of your overall grade). It is due Monday, September 21st before midnight.

The Problem

You will use Turtle graphics to draw a congruent pentagon (5 sided figure, each side of the same length) centered on the origin of a 2D graphics window with a fill color given by the user. The user provides three floating point values indicating the proportions of the colors red, green and. You will also label the drawn pentagon.

There are many ways to create a color but a common one used in computer graphics is the process of additive color (see http://en.wikipedia.org/wiki/Additive_color). Combining different amounts of red, green and blue can create most, but not all, colors.

Originally written as a part of the *logo* programming language, Turtle graphics (http://en.wikipedia.org/wiki/Turtle_graphics) is one of the oldest graphics programs. It 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. Python has a module that implements the behavior of the original turtle graphics program and this module is simply called "turtle".

Your program will read in color values from the user and then use the methods in the turtle module to draw a pentagon centered on the origin filled with the color indicated. You will also then create a label for the pentagon.

Program Specifications

Your program will function as follows:

1. output a brief, descriptive message when it first starts, indicating the purpose of the program and the user-required input and output that will be provided.
2. prompt the user for the three color values. Each value is a floating point number between 0 and 1.0
3. prompt the user for the length of one side of the pentagon.
4. draw a congruent pentagon centered on the origin filled with the indicated color
5. the pencolor should be set to the **inverse** of the provided fill color. If the fill color is (1.0, 0.6, 0.4), then the pen color should be (0.0, 0.4, 0.6).
6. provide a caption underneath the pentagon indicating the three color values used for the color of the pentagon

Deliverables

proj02.py -- your source code solution (remember to include your section, the date, project number and comments).

1. Please be sure to use the specified file name, i.e. "proj02.py"
2. Save a copy of your file in your CS account disk space (H: drive on CS computers).
3. Electronically submit a copy of the file.

Assignment Notes:

Using turtle graphics:
In order to use turtle graphics in python you must first import the turtle module. You can then use the help function in IDLE to find out what methods this module includes and what they do. Just type `import turtle` in the IDLE command window, hit enter, and then type `help(turtle)` and scroll up through the list and information.

The basic concept behind the turtle is the pen. The pen is either up or down. When down, the turtle draws as it moves around the Cartesian graph. There are a number of methods that are needed in this project:

`turtle.up(),turtle.down():` Set the pen state to be up (not drawing) or down (drawing)

`turtle.right(degrees), turtle.left(degrees):` Turn the direction that the turtle is facing. The amount of turn is indicated in degrees.

`turtle.forward(distance), turtle.backward(distance):` Move the turtle forward or backward the amount of distance indicated. Depends on the direction the turtle is facing. Draws a line if the pen is down, not if the pen is up.

`turtle.goto(x,y):` The goto method moves the turtle to a specified point, drawing a line along the way if the pen is down, and not drawing if the pen is up. Note: The turtle always starts at the point (0,0). The goto method moves to the indicated x,y coordinates.

`turtle.fillcolor(r,g,b):` The fillcolor method sets the color for filled figures until the color is changed. It takes three arguments, each a floating point number between 0.0-1.0. The first is the amount of red, the second, the amount of green and the third the amount of blue.

`turtle.pencolor(r,g,b):` The pencolor method sets the color of the pen until the color is changed. It takes three arguments, each a floating point number between 0.0-1.0. The first is the amount of red, the second, the amount of green and the third the amount of blue.

`turtle.pensize(width):` The pen is set to a width passed as a parameter (integer)

`turtle.write(string):` Write a string starting at the present turtle point.

`turtle.begin_fill():` Figures drawn subsequently will be filled with the `fillcolor` set above.

`turtle.end_fill():` Filling of figures is halted. Figures drawn subsequently will no longer be filled with the `fillcolor`.

First, try it!
The first thing you should do is try out some of the commands by just typing them in the idle window. You can get a much better feel for how the whole thing works by just trying it.

Making a string
You can make a string using concatenation (the + operator) and the str() function. Two strings can be concatenated by "adding" them. Thus

"bill" + "fred"

yields the string "billfred". If you want to change a value to a string, use the str() function. Thus

"bill" + str(4)

yields the string "bill4". This will be helpful for making the caption.

Keeping the window up
The drawing window has a tendency to disappear very quickly. One way to "hold" the window is to use the sleep function in the time module, as follows:

```
import time
time.sleep(seconds)
```

The program will wait the number of seconds indicated before it ends.

Working nicely with IDLE
IDLE does not tend to work well with other graphics windows. Subsequently, after you open a progam in IDLE that does graphics (such as turtle), some of the windows may hang, or not close. To get around this, you can use the _exit function in the os module. Note the underline in the name. Make the _exit funtion the last line in the program.

```
import os
os._exit(1)
```

Getting Started

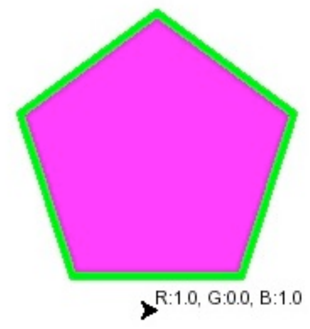
1. Use IDLE create a new program.
2. If you are in a CSE lab, select the H: drive as the location to store your file
3. Save the name of the project: proj02.py
4. Create the preface print information and prompt for the user input (4 separate prompts)
5. Run the program and fix any errors
6. Use the web site to hand in the program (incomplete as this point but you should continually hand things in)
7. First see if you can just draw a circle. How do you get it to be centered on the origin?
8. Next, set any color and see if you can draw the circle again
9. Now, see if you can fill the circle with a color
10. Finally, get the prompt in for the three colors and draw the circle.

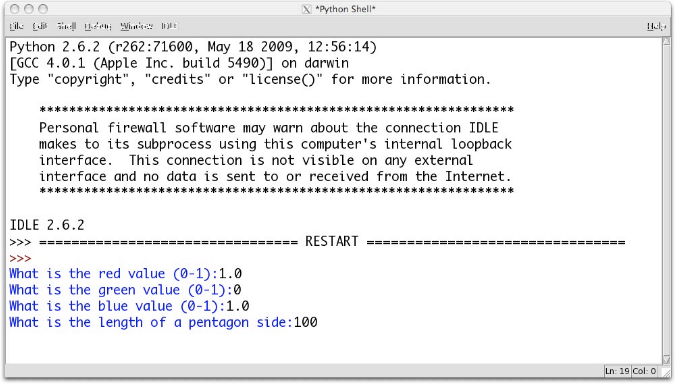
11. Now you enter a cycle of edit-run to incrementally develop your program.
12. Handin to hand in your final version.

Questions for you to consider (not hand in)

1. What combination of r,g,b will produce: white, black, cyan, other colors?
2. Can you do the same thing by drawing a square instead of a circle (there is no “square” command).

Example output





```
Python 2.6.2 (r262:71600, May 18 2009, 12:56:14)
[GCC 4.0.1 (Apple Inc. build 5490)] on darwin
Type "copyright", "credits" or "license()" for more information.

    ****************************************************************
    Personal firewall software may warn about the connection IDLE
    makes to its subprocess using this computer's internal loopback
    interface.  This connection is not visible on any external
    interface and no data is sent to or received from the Internet.
    ****************************************************************

IDLE 2.6.2
>>> ================================ RESTART ================================
>>> 
What is the red value (0-1):1.0
What is the green value (0-1):0
What is the blue value (0-1):1.0
What is the length of a pentagon side:100
```