

Programming Project 10

This assignment is worth 50 points and must be **completed and turned in before 11:59 on Wednesday, 04/14/ 2010.**

Assignment Overview

This assignment will give you more experience on the use of:

1. classes
2. class methods

In this project, we are going to use a library package called turtle graphics to draw some pictures, but we will define and use classes to create the basic shapes and assemble them into a picture. You are going to create at least 5 classes. Using instances of these classes, you are going to draw a simple scene. You will write a function that assembles the scene.

Project Description / Specification

1. Define 5 classes (at least). One can be a circle class and one can be a rectangle class (which are both easy), but you must come up with at least 3 “interesting” classes as well (not simple turtle commands, but some construction of multiple commands to draw the class instance)
2. Each class will have an `__init__`, `__str__` and a `draw` method:
 - a. the `init` method will take a string argument, either indicating a fill color or, if the argument is “” (the empty string), that the shape is not filled.
 - b. the `draw` method will take at least x and y coordinates arguments (indicating where the figure is drawn) and a `turtle.Turtle` object to use for drawing the shape.
 - c. the `str` is the conversion to a string; it returns the string to be used for printing in python.
 - d. Other arguments may be required for your methods; all arguments will be described in your docstrings.
3. a function `scene(pen, time=True)` where `pen` is a `turtle.Turtle` object with which to draw the scene and `time` indicates if the scene is a day-time scene (`True`) or night-time scene (`False`). Your scene must, at a minimum, include:
 - a. a house, with at least one door and two windows
 - b. a tree
 - c. the sun, if a day-time scene, or the moon, if a night-time scene
 - d. night-time colors should be more subdued than day-time colors
4. All classes, methods and functions **require** a docstring for a general description of the object/method/function.

Deliverables

Turn in `proj10.py` containing all of your class and function definitions.

1. Please be sure to use the specified file name, i.e. “`proj10.py`”

2. Save a copy of your file in your CSE account disk space (H drive on CSE computers).
3. Submit the files using the "handin" program:
<http://www.cse.msu.edu/handin/webclient>

Assignment Notes:

The idea is to make classes for objects in your scene (house(s), windows, door(s), tree(s), whatever) that have to be drawn and are not just a circle or a square/rectangle.

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. Assuming the assignment `pen = turtle.Turtle()`, some useful features include,:

- `pen.up()`, `pen.down()`: Set the pen state to be up (not drawing) or down (drawing)
- `pen.right(degrees)`, `pen.left(degrees)`: Turn the direction that the pen is facing. The amount of turn is indicated in degrees.
- `pen.forward(distance)`, `pen.backward(distance)`: Move the pen forward or backward the amount of distance indicated. Depends on the direction the pen is facing. Draws a line if the pen is down, not if the pen is up.
- `pen.goto(x,y)`: Move the pen to the specified point, drawing a line along the way if the pen is down, and not drawing if the pen is up.
- `pen.pencolor(r,g,b)`, `pen.pencolor(s)`: Set the color that the pen will hold for all drawing until the pen color is changed. In the first version, each argument is 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. In the second version, the argument is a string indicating a color by name or by its hex code, e.g., "green", "red", "#66FFFF". Hex color codes are at: [//www.web-source.net/216_color_chart.htm](http://www.web-source.net/216_color_chart.htm)
- `pen.fillcolor(r, g, b)`, `pen.fillcolor(s)`: Set the color for filling figures. Arguments are the same as for the pen color.
- `pen.circle(radius)`: draw a circle of the indicated radius. The circle is drawn tangent to the direction of the pen in a clockwise direction (if radius is positive).
- `pen.write(string)`: Write a string starting at the present pen point.
`pen.fill(flag)`: To fill a figure, use the command `pen.fill(True)` before you start drawing it. Draw the figure, then execute the command `pen.fill(False)`. The figure drawn between the two fill commands will be filled with the present color setting.
- `pen.clear()`: Clear (erase) everything written by the pen.

The “proj10.py” file that you hand in must contain your class and function definitions. The TA will test your program by running the test program “test_proj10.py”, which calls your scene function.

The file “turtle_example.py” is a simple example illustrating how to use turtle graphics. Bring up a python shell, load this example program (which imports turtle), then create a pen and call the functions defined in the example program with different size arguments. For instance, type in the shell:

```
pen = turtle.Turtle()
drawSquare(pen, fillcolor="green")
pen.clear()
drawCircles(pen, 30)
drawCircles(pen, num=12)
```

Continue to learn about turtle graphics by invoking other methods on `pen`.

The pictures below illustrate a daytime scene and a nighttime scene that meet the requirements.

