Lab Exercise 9

Software Design **Allen B. Downey**
Spring 2006

Due: never!!!

9.1 GUI

1. Download the latest versions of `Gui.py` and `Lumpy.py`. If you use `wget`, remember to delete the old one first.

2. Download widget_demo.py and `HelloGui.py` from the usual place and run them. Read the code and make sure you understand how it works. widget_demo.py demonstrates the basic use of widgets; `HelloGui.py` demonstrates the usual structure of an interface that inherits from `Gui`.

   It also displays an object diagram and a class diagram that demonstrate the relationships among the objects and classes you will be working with. The class diagram is a mess because Lumpy has a dump graph layout algorithm, but I will hand out a cleaner version.

3. Download `MyWorld.py` from the usual place. It is an example of how you can use inheritance to customize the behavior of existing classes without having to modify the original. This module contains three class definitions:

   MyWorld: extends `TurtleWorld` and overrides `setup`, allowing you to customize the Gui features for the World.

   MyTurtle: extends `Turtle`, although at the moment it doesn't override anything.

   MyTurtleControl: extends `TurtleControl` and overrides `setup`, again so you can customize the Gui features for the turtle.

   As usual, you should read over the code and make sure you understand it.

4. The purpose of this lab exercise is to encourage you to play around and get familiar with the GUI elements in Tkinter. You might find is useful to try the following exercises, although you don't have to do them all.

   You can also experiment with rearranging the widgets, but for now you probably don't want to spend much time adjusting their layout or appearance. We will play with the geometry manager more next week.

   (a) Add a button to `MyWorld.setup` that draws an item on the canvas when it is pressed.

   (b) Copy the `draw` method from `Turtle` into `MyTurtle` and then modify it so that your Turtles look different from those plain-belly turtles.

   (c) Add a new method to the `MyTurtle` class, like `dance`, and then add a button to the `MyTurtleControl` panel so that when the button is pressed, it invokes the new Turtle method.

(d) While you are at it, modify your Turtles so that their "pen color" is their current color.

(e) Add an entry to the Turtle control panel and put the integer 1 into it. Then modify the Turtle so that when it moves, it draws a line with the line width specified in the entry.

(f) Add up and down buttons that increment and decrement the contents of the line width entry.

(g) Replace the pen up and pen down buttons with a single check button that controls whether the pen is down.

(h) Replace the menubutton that controls the turtle color with a sequence of checkbuttons or a listbox.