Lab Exercise 9

Software Design **Allen B. Downey**
Fall 2004

Due: never!!!

9.1 GUI

1. If you haven't already, download `HelloGui.py` and run it. Review the code and draw the widget hierarchy it builds.

2. The first argument for each widget function controls the arrangement of widgets. The only two values we will use are `LEFT` for widgets that are arranged left-to-right, and `TOP` for widgets that are arranged top-to-bottom. The default is TOP, so you can leave it out (but you probably shouldn't).

   Rearrange the contents of the window so that the buttons are on the left and the canvas is on the right. Now rearrange again so that the buttons are on the bottom.

3. The parameters `fill` and `expand` are confusing, but if you play around with them, you will figure it out. The general idea is that each widget is packed into an imaginary region that is determined by the size and arrangement of the other widgets. Most widgets have a minimum size, but many widgets can be stretched if desired. The `fill` parameter can be `X`, `Y`, `BOTH` or `NONE`. It specifies which direction(s) the widget should fill. Confusingly, "filling" a space doesn't necessarily mean that the widget will look bigger. If you want the widget to expand to fill the allocated space, you have to include `expand=1`.

   To see how this works, see if you can figure out how to make the buttons "fill" their frame and "expand" to the same width as the canvas (or height if they are still beside the canvas). Hint: you will have to expand the buttons *and* the frame that contains them.

4. Review `Gui.create_text` and see if you can figure out how to change the text color.

5. Add a button with the label "Circle" that invokes a method named `circle` that creates a circle with radius 100 in the center of the canvas. It should take no parameters (except `self`), so you don't have to make a `Callable` object.

6. Add another button that creates a rectangle. Notice that the `create_rectangle` method expects a list of two coordinates, where each coordinate is a list of two numbers. The coordinates specify the upper-left and lower-right corners of the rectangle. You should use a `Callable` object to encapsulate `create_rectangle` and its parameters.

7. Bring up the Tkinter documentation in a browser:

   `http://www.python.org/topics/tkinter/doc.html`

   In particular, check out the documentation of the `Button` widget:

`http://www.pythonware.com/library/tkinter/introduction/button.htm`

Change the foreground or background color of one of the buttons. Try changing the background color of the canvas.

8. The easiest way to specify a font is with a tuple that contains a font family name (string) and a size (int). Try this:

```
def hello(self):
    self.create_text([0, 0], 'Hello', font=('Helvetica', 36))
```

9. Sometimes to get the documentation you need, you have to drop down a level and check the Tk documentation. For example, to look up the options for a text item on a `Canvas`, check out:

   `http://www.tcl.tk/man/tcl8.5/TkCmd/canvas.htm`

   You should bookmark a lot of these pages, and print the ones you find yourself using frequently. Also, for example of how to use various widget and items, you can look in `World.py`.

10. Read the documentation of `Entry` in the Tkinter docs, then look in `World.py` and check out `TurtleControl.setup` and `TurtleControl.fd`. Then add a new button to `HelloGui` with the label "Display." Add an Entry next to the button, so that when the button is pressed, it takes the contents of the entry and displays them as a text item in the canvas.

I hope this helps you get started with GUIs! If you have a hard time, I suggest that you do what you can in a reasonable amount of time, and then stop. I will distribute a solution next week.