GUI Programming with Python: Tkinter

The tkinter package (“Tk interface”) is the standard Python interface to the Tk GUI toolkit. Both Tk and tkinter are available on most Unix platforms, as well as on Windows systems. (Tk itself is not part of Python; it is maintained at ActiveState.)

Tk provides the following widgets:

- button
- canvas
- checkbutton
- combobox
- entry
- frame
- label
- labelframe
- listbox
- menu
- menubutton
- message
- notebook
- tk_optionMenu
- panedwindow
- progressbar
- radiobutton
- scale
- scrollbar
- separator
- sizegrip
- spinbox
- text
- treeview

It provides the following top-level windows:

- tk_chooseColor - pops up a dialog box for the user to select a color.
- tk_chooseDirectory - pops up a dialog box for the user to select a directory.
- tk_dialog - creates a modal dialog and waits for a response.
- tk_getOpenFile - pops up a dialog box for the user to select a file to open.
- tk_getSaveFile - pops up a dialog box for the user to select a file to save.
- tk_messageBox - pops up a message window and waits for a user response.
- tk_popup - posts a popup menu.
- toplevel - creates and manipulates toplevel widgets.

Tk also provides three geometry managers:

- pack - which packs widgets into a cavity
- place - which positions widgets at absolute locations
- grid - which arranges widgets in a grid

Simple Tkinter program

```
from tkinter import *
counter = 0
def counter_label(label):
    def count():
        global counter
        counter += 1
         label.config(text=str(counter))
        label.after(1000, count)
    count()
root = Tk()
root.title("Counting Seconds")
label = Label(root, fg="green")
label.pack()
button = Button(root, text='Stop', width=25, command=root.destroy)
button.pack()
counter_label(label)
root.mainloop()
```

Lab 1:

Change the program such that every second the button displays a different label, instead of “stop”.

Intermediate Tkinter program

```
from tkinter import *
import random

root = Tk()
root.title("Simple Graph")
root.resizable(0,0)
points = []
backgrounds = ["#ff998A", "#ff0000", "#00ff00", "#0000ff"]

def point(event):
            c.create_oval(event.x, event.y, event.x+1, event.y+1, fill="black")
            points.append(event.x)
            points.append(event.y)
            print(points)
            return points

def graph(event):
            c.create_line(points)

def changeBg():
        c.configure(background = backgrounds[random.randrange(0, len(backgrounds))])

c = Canvas(root, bg="white", width=300, height= 300)
c.pack()
button = Button(root, text='change color', width=25, command=changeBg)
button.configure(cursor="fleur")
button.pack()

root.configure(cursor="crosshair")
root.bind("<Button-1>", point)
root.bind("<Double-1>", graph)
root.mainloop()
```

Lab 2 / Homework:

1. Add a couple of buttons to your GUI program
   - One button should close (destroy) the window;
   - Another button should create a square and fill it with yellow color. See http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/create_rectangle.html for reference on how to create a rectangle.
2. Add something else to your program – your imagination is the limit, constrained only by the TKinter capabilities: http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/index.html