EGR 680
Fall 2018
Laboratory #8

Graphical User Interface (GUI) in Python

Due Date: By beginning of next lab section

Objectives

- To be able to apply your knowledge of Python classes in a more advanced application
- Integrate multiple classes and methods into your design
- To be able to generate a graphical user interface (GUI)

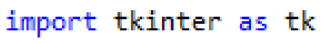
Part I – Simple GUI

In this part, you will create a simple Python GUI using the popular ***Tkinter*** GUI library. The ***Tkinter*** library is event driven in nature and can respond to many different types of events including key press, key release, button press, button release, motion and many more types of events. In this part, you will only be exploring the button press event. This library comes packaged with the Anaconda distribution, so there is no need to install any additional libraries. Your GUI will launch in a pop-up window outside of Spyder. Before creating your GUI, investigate the ***Tkinter*** library and see what methods are available and understand how the methods work.

1. Launch the **Anaconda** distribution and then launch **Spyder**.

2. Create a new file and save it as ***MyGUI.py***.

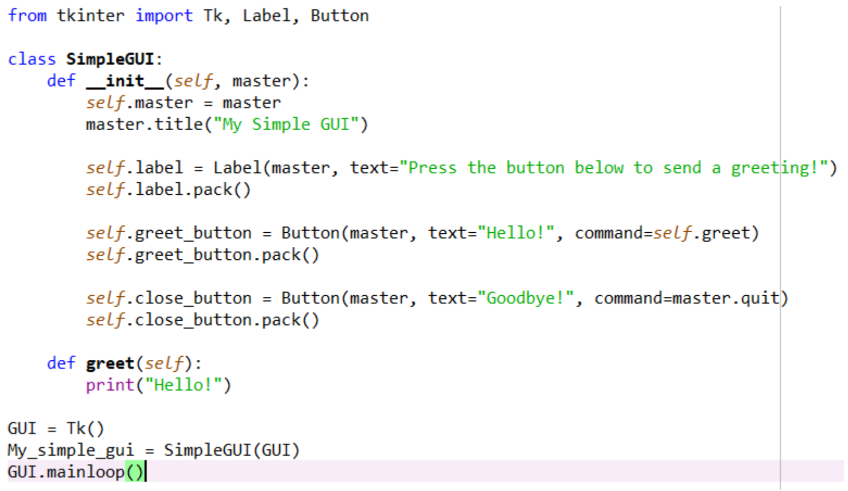
3. To start, let’s first investigate the ***Tkinter*** library. Type the following code into your **Python** file:



```
import tkinter as tk
```

4. With your cursor somewhere on the word ***Tkinter***, press **CTRL+G**. This will open the ***Tkintner*** library. You can do this with any imported library.

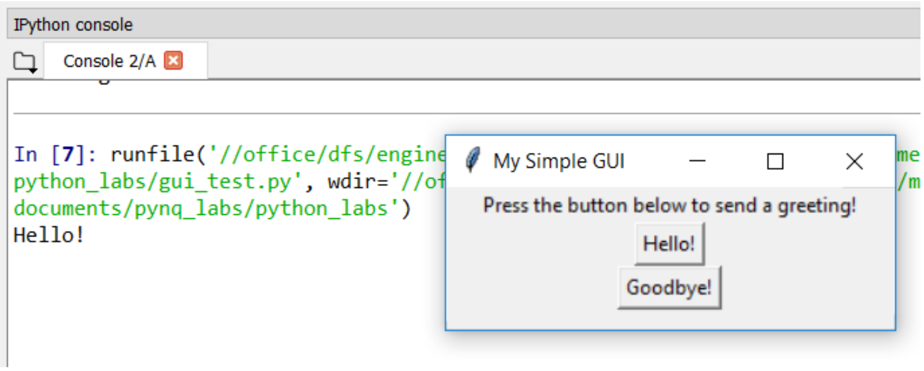
5. Copy the code below to create a simple GUI.



```
from tkinter import Tk, Label, Button

class SimpleGUI:
    def __init__(self, master):
        self.master = master
        master.title("My Simple GUI")

        self.label = Label(master, text="Press the button below to send a greeting!")
        self.label.pack()

        self.greet_button = Button(master, text="Hello!", command=self.greet)
        self.greet_button.pack()

        self.close_button = Button(master, text="Goodbye!", command=master.quit)
        self.close_button.pack()

    def greet(self):
        print("Hello!")

GUI = Tk()
My_simple_gui = SimpleGUI(GUI)
GUI.mainloop()
```

Figure 1: A simple GUI

6. A window should appear with your GUI when you **Run** the program. When you press the ***Hello!*** button, you should see your greeting in the console. When you press the ***Goodbye!*** button, your program should exit.



```
In [7]: runfile('//office/dfs/engine
python_labs/gui_test.py', wdir='//of
documents/pynq_labs/python_labs')
Hello!
```

Press the button below to send a greeting!

Figure 2: GUI output

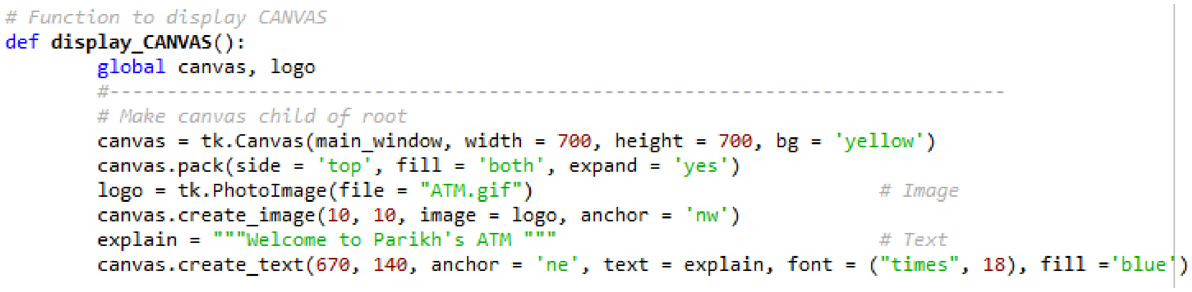
7. Pick at least one new function from the ***Tkinter*** library, internet resources and lecture slides and implement a new feature in your GUI.

Part II – ATM GUI

In this part, you will create a GUI for the ATM machine program you designed in Lab 7. Recreate the ATM program using functions and a GUI. Your ATM machine should display a static image and a welcome message at the top of the window. The functionality of the ATM should be the same as before. Remember to look through the ***Tkinter*** library for different ways to implement the GUI.

1. Create a new **Python** file and save it as ***ATM_GUI.py***.

2. Use the sample code in Figure 3 as a starting point for your program. This code uses the ***canvas*** class to generate the background for the ATM machine.

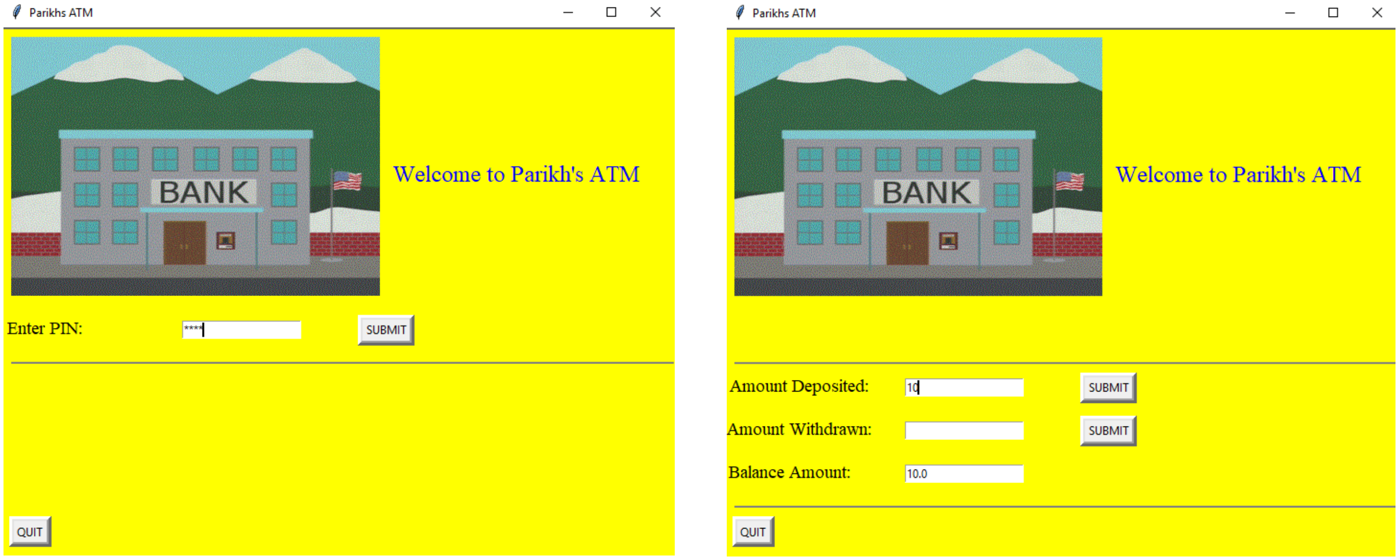


```
# Function to display CANVAS
def display_CANVAS():
        global canvas, logo
        #---------------------------------------------------------------------------
        # Make canvas child of root
        canvas = tk.Canvas(main_window, width = 700, height = 700, bg = 'yellow')
        canvas.pack(side = 'top', fill = 'both', expand = 'yes')
        logo = tk.PhotoImage(file = "ATM.gif")                          # Image
        canvas.create_image(10, 10, image = logo, anchor = 'nw')
        explain = """Welcome to Parikh's ATM """                        # Text
        canvas.create_text(670, 140, anchor = 'ne', text = explain, font = ("times", 18), fill ='blue')
```

Figure 3: GUI background code

3. Design the rest of the GUI while implementing all of the features from Lab 7.

4. Figure 4 shows an example of how your GUI should work. Do not copy the design of this GUI. Make it your own!



Welcome to Parikh's ATM

Enter PIN: ****

Welcome to Parikh's ATM

Amount Deposited: 10

Amount Withdrawn:

Balance Amount: 10.0

Figure 4: Final GUI design

Part III – ATM GUI using Classes

Modify your program from Part II to utilize classes and methods instead of functions. This part should not require a significant amount of effort, but it is a necessary step moving forward. Next week's lab will be quite challenging, so the more prepared you are with Python, the better.

1. Create a new **Python** file and save it as ***ATM_GUI_class.py***.

2. There should be no difference in functionality between Part II and Part III. The end user should not be able to see any difference between the two GUIs.

***Extra Credit*: Upon exit of the ATM machine, print a receipt that provides detailed transaction activity.**

Laboratory Deliverables

You are required to turn in a hard copy of the report. Report should have the following items:

- Cover page with one page description of your design
- Printouts of your code for Part II and Part III (with comments)
- Snapshot of sample run of your output

You also have to demonstrate your design and turn in code for Parts II and III.