CSC 157

Name \_James Aniciete\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date \_4/3/2020\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lab No. \_\_\_08\_\_\_\_

Purpose of the Lab Activity

**The purpose of this lab activity is to modify a GUI program (tkContacts.py) that will allow one to add, update, delete, and load contacts from a list defined in another py file (contacts.py).**

Source Code

from tkinter import \*

from contacts import \*

def selection () :

print ("At %s of %d" % (select.curselection(), len(contactlist)))

return int(select.curselection()[0])

def addContact () :

contactlist.append ([nameVar.get(), phoneVar.get()])

setList ()

def updateContact() :

contactlist[selection()]=[nameVar.get(), phoneVar.get()]

setList ()

def deleteContact() :

del contactlist[selection()]

setList ()

def loadContact () :

name, phone = contactlist[selection()]

nameVar.set(name)

phoneVar.set(phone)

def buildFrame () :

global nameVar, phoneVar, select

root = Tk()

frame1 = Frame(root)

frame1.pack()

Label(frame1, text="Name:").grid(row=0, column=0, sticky=W)

nameVar = StringVar()

name = Entry(frame1, textvariable=nameVar)

name.grid(row=0, column=1, sticky=W)

Label(frame1, text="Phone:").grid(row=1, column=0, sticky=W)

phoneVar= StringVar()

phone= Entry(frame1, textvariable=phoneVar)

phone.grid(row=1, column=1, sticky=W)

frame1 = Frame(root) # add a row of buttons

frame1.pack()

btn1 = Button(frame1,text=" Add ",command=addContact)

btn2 = Button(frame1,text="Update",command=updateContact)

btn3 = Button(frame1,text="Delete",command=deleteContact)

btn4 = Button(frame1,text=" Load ",command=loadContact)

btn1.pack(side=LEFT); btn2.pack(side=LEFT)

btn3.pack(side=LEFT); btn4.pack(side=LEFT)

frame1 = Frame(root) # allow for selection of names

frame1.pack()

scroll = Scrollbar(frame1, orient=VERTICAL)

select = Listbox(frame1, yscrollcommand=scroll.set, height=7)

scroll.config (command=select.yview)

scroll.pack(side=RIGHT, fill=Y)

select.pack(side=LEFT, fill=BOTH)

return root

def setList () :

contactlist.sort()

select.delete(0,END)

for name,phone in contactlist :

select.insert (END, name)

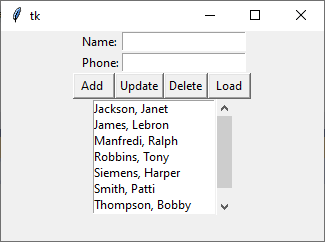
root = buildFrame()

setList ()

root.mainloop()

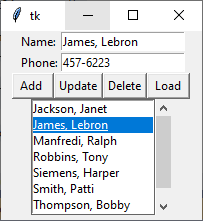
Snippet(s) of Output(s)

**Initial Display:**

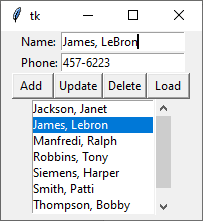


**Update – correct LeBron’s name:**

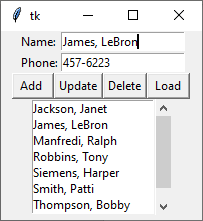
* Load his name and phone number



* Modify name

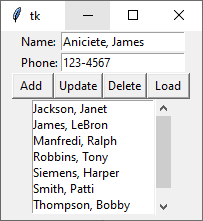


* Click Update button

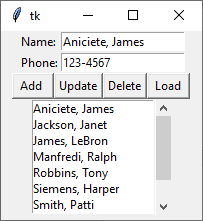


**Add James Aniciete with 123-4567**

* Enter contact information

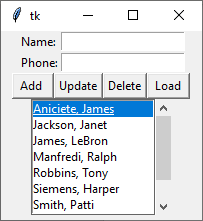


* Click Add button

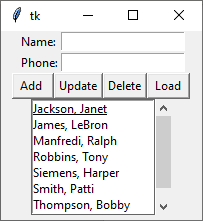


**Delete Aniciete contact**

* Click Aniciete contact in ListBox



* Click Delete button



Modified Source Code

# Programmer: James Aniciete

# Course No.: CSC 157

# Lab No.: 08 - GUI Programming with Tkinter

from tkinter import \*

from tkinter import messagebox # for exit button's messagebox

from contacts import \*

import os # for exiting the app

# Prints the ListBox selection's index and the length of the contact list

def selection () :

print ("At %s of %d" % (select.curselection(), len(contactlist)))

# returns index of a tuple

return int(select.curselection()[0])

# adds a new contact to the contact list

def addContact () :

contactlist.append ([nameVar.get(), phoneVar.get()])

setList ()

# update selected contact in the contact list w/ the nameVar and phoneVar

def updateContact() :

contactlist[selection()]=[nameVar.get(), phoneVar.get()]

setList ()

# delete selected contact from the contact list

def deleteContact() :

del contactlist[selection()]

setList ()

# load the selected contact's data into the name & phone variables

def loadContact () :

# put name and phone selections into a tuple

name, phone = contactlist[selection()]

# use tuple to assign values to name and phone variables

nameVar.set(name)

phoneVar.set(phone)

# function that places all widgets into the frame individually

def buildFrame () :

# define global variables

global nameVar, phoneVar, select

# create the main window widget

root = Tk()

# add title to the frame

root.title("My Contact List")

# create & pack a frame in the root window

frame1 = Frame(root)

frame1.pack()

# on 1st row of frame:

# create a label for name

Label(frame1, text="Name:").grid(row=0, column=0, sticky=W)

# initialize StringVar for name

nameVar = StringVar()

# assign entry button value to the name var

name = Entry(frame1, textvariable=nameVar)

# position name var in first row, second column, aligned to the west cell border

name.grid(row=0, column=1, sticky=W)

# on 2nd row of the frame:

# create a label for phone no.

Label(frame1, text="Phone:").grid(row=1, column=0, sticky=W)

# create string var for phone no.

phoneVar= StringVar()

# assign entry button value to phone var

phone= Entry(frame1, textvariable=phoneVar)

# position phone var in second row, second column, aligned to the west

phone.grid(row=1, column=1, sticky=W)

# create & pack a frame in the root window

frame1 = Frame(root)

frame1.pack()

# add a row of buttons to frame1 with respective callback functions

btn1 = Button(frame1,text=" Add ",command=addContact)

btn2 = Button(frame1,text="Update",command=updateContact)

btn3 = Button(frame1,text="Delete",command=deleteContact)

btn4 = Button(frame1,text=" Load ",command=loadContact)

btn5 = Button(frame1,text=" Save ",command=saveContact)

# pack the buttons on the same row to the left

btn1.pack(side=LEFT)

btn2.pack(side=LEFT)

btn3.pack(side=LEFT)

btn4.pack(side=LEFT)

btn5.pack(side=LEFT)

# allow for selection of names from a ListBox with a scrollbar

frame1 = Frame(root)

frame1.pack()

# create a vertical bar widget

scroll = Scrollbar(frame1, orient=VERTICAL)

# whichever value from the ListBox is clicked is assigned to select

# height = # of values visible in the Listbox

select = Listbox(frame1, yscrollcommand=scroll.set, height=7)

scroll.config (command=select.yview)

scroll.pack(side=RIGHT, fill=Y)

select.pack(side=LEFT, fill=BOTH)

# create frame for Exit button at the bottom of the window

frame2 = Frame(root)

frame2.pack()

# create exit button & pack it

btn6 = Button(frame2, text = " Exit ", command = exitContact)

btn6.pack()

# return root object to allow for the frame to be built

return root

# sorts the contact list & allows for an update to the ListBox

def setList () :

contactlist.sort()

select.delete(0,END)

for name,phone in contactlist :

select.insert (END, name)

# save the updated contactList

def saveContact():

fobj = open("contacts.py", "w")

fobj.write("contactlist = [")

for items in contactlist:

fobj.write(str(items))

fobj.write(",")

fobj.write("\n")

fobj.write("]")

# display MessageBox entitled Contacts, providing option to exit the app

def exitContact():

app\_title = "Contacts"

if messagebox.askokcancel(title = app\_title, message = "Do you want to exit, OK or Cancel") == 1:

os.\_exit(1)

root = buildFrame()

setList ()

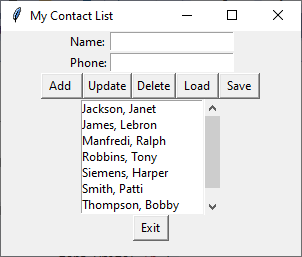
# set size of window (width x height)

root.geometry("300x225")

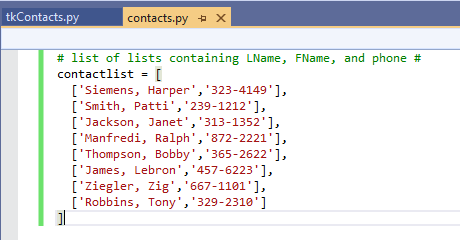
root.mainloop()

Snippet(s) of Output(s) from execution of modified Code

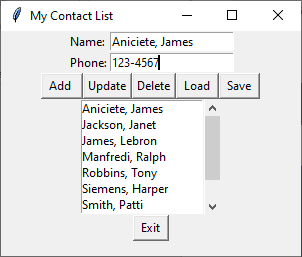
- Entire interface at runtime including save & exit buttons



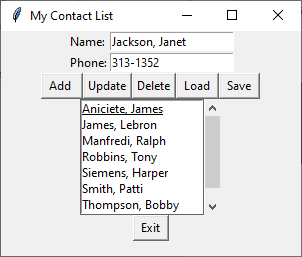
-Initial snapshot of contacts.py at runtime



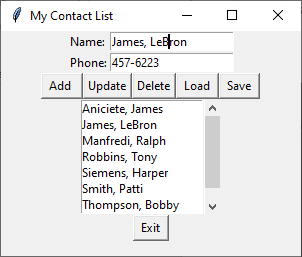
-Show a contact being added (James Aniciete, 123-4567)



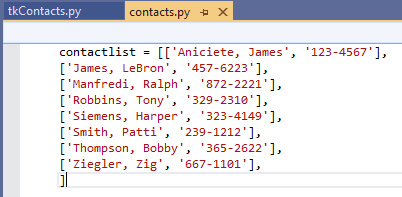
-Show a contact deleted (Janet Jackson)



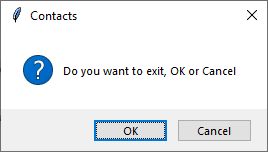
-Show a contact being updated (Lebron 🡺 LeBron)



-Show modified contactlist file



-Your message box requesting if a user wants to exit the program



Excel Spreadsheet (when Calculations are involved)

n/a

Answers to Questions (Be sure to copy the questios themselves!)

**(1)** What is Tkinter? How is it similar to Turtle Graphics?

**Tkinter is a module that allows you to create simple GUI programs. Tkinter is similar to Turtle Graphics in that you are able to draw images in both; however, Tkinter is more complex in that one can add buttons, labels, list boxes, etc.**

**(2)** When coding GUI applications, how does a radio button differ from a check box?

Hint: visit the following Web site for additional information

<https://www.tutorialspoint.com/python/python_gui_programming.htm>

**A radio button is used for selecting one option from a list of mutually exclusive options while a check box is used for selecting either On/Off or Yes/No options.**

**(3)** What was the purpose of the [ Load ] button that was used in your application?

**The purpose of the [Load] button is to load the selected contact’s data into the variables nameVar and phoneVar, assigning the selected data to their respective variable.**

**(4)** How are text files used in your program code for this project?

**A text file called contacts.py is used to store the initial contact list. Then, the Save button can be used to overwrite the file with an updated contact list based on whatever modifications were made using the GUI interface.**

**(5)** What have you learned from performing and coding for this lab assignment?

**I have learned how to build GUI frames and how to work with callback functions. I am still unsure of why frame1 was both initialized and packed repeatedly. Could the other initializations just use a different frame number? I feel like there were 4 frames/sections of the window, but the code only used 2.**