Python Lab Assignment 7

# Program description

Implement a GUI with Python's **Tkinter** module for the given image. Use the ***names.txt***text file to generate names (first 10 names) and list inside the name generator GUI. Also generate a Name based on the alphabet selection through the **Combobox** using random function.  
  
**Note 1**: For all these use the given names.txt file. Try to create GUI as similarly as possible.  
**Note 2**: Do this on your own. Demo files are already posted in the theory classroom.  
**Note 3**: Use your register number and name as the title of the window.

# Code

## Non-GUI support functions module

from random import choice

**def** trim(s):

i = 0

if s.isspace(): return " " *# If the whole string is full of spaces.*

*# Removing leading spaces...*

while s[i].isspace(): i = i + 1

s, i = s[i:], -1

*# Removing trailing spaces...*

while s[i].isspace(): i = i - 1

if i != -1: s = s[:(i + 1)]

return s

**def** getTopNames(c, n):

names, c, i = "", c.upper(), 0 *# N => selected names*

*#----------------------*

*# Getting n names starting from letter denoted by c.*

f = open("names.txt", "r")

*# We assume every line is a name, and hence every name ends with a newline character.*

name = f.readline()

while name != "" and i < n:

if name[0].upper() == c: names, i = names + name, i + 1

name = f.readline()

f.close()

return trim(names)

**def** getRandomName(c):

N, c = [], c.upper() *# N => selected names*

*#----------------------*

*# Getting all names starting from letter denoted by c...*

*#------*

f = open("names.txt", "r")

*# We assume every line is a name, and hence every name ends with a newline character or is at the end of the file.*

name = f.readline()

while name != "":

if name[0].upper() == c: N.append(name)

name = f.readline()

*#--------------------*

*# Returning a random name...*

*#------*

f.close()

return trim(choice(N))

**def** alphabetList(): return tuple(map(chr, range(ord("A"), ord("Z") + 1)))

## GUI support functions

from tkinter import \*

**def** getDim(s):

lines = s.split("\n")

width = max(list(map(len, lines)))

height = len(lines)

return (width, height)

**def** updateTextBox(t, s, flex, start, end):

*# t => text box widget*

*# s => string to be inserted*

*# flex => flexible dimensions, based on the string*

if flex: (t["width"], t["height"]) = getDim(s)

t["state"] = "normal"

t.delete(start, end)

t.insert(start, s)

t["state"] = "disabled"

**def** utb\_simple(t, s): updateTextBox(t, s, False, "1.0", "end")

**def** utb\_flex(t, s): updateTextBox(t, s, True, "1.0", "end")

**def** utb\_header(t, s): updateTextBox(t, s, False, "2.0", "end")

*# "end-1c" leaves the newline character of the header.*

## Main program

from support\_gui import \*

from support\_nongui import \*

from tkinter.ttk import Combobox

*#========================*

*# MAIN WINDOW*

w = Tk()

w.geometry("500x450")

w.title("1940223 - Pranav Gopalkrishna")

*#========================*

*# FRAMES*

frames = [

Frame(w, pady = 20),

Frame(w, padx = 20, pady = 20),

Frame(w, padx = 20, pady = 20)]

*# Packing frames...*

for f in frames: f.pack(side = "top")

*#========================*

*# INPUT COMBO BOX*

promptLabel = Label(

frames[0],

text = "Starting alphabet",

font = ("Arial", 15))

inp = Combobox(frames[0], values = alphabetList(), width = 2)

*#========================*

*# RANDOM NAME GENERATION*

*# Random name generating function and trigger button...*

display = Text(

frames[1],

height = 1,

font = ("Arial", 15),

state = "disabled",

borderwidth = 2,

relief = "ridge",

padx = 10,

pady = 10)

**def** displayRandomName():

c = inp.get()

if c.isalpha(): utb\_flex(display, "A random name from \"" + c + "\" is " + getRandomName(c) + ".")

else: utb\_flex(display, "Input alphabets only!")

utb\_flex(display, "Generated name will be displayed here.") *# Initial default text.*

generateButton = Button(frames[0], text = "GENERATE", command = displayRandomName)

*#========================*

*# TEXT BOXES*

texts, fa = [], ("M", "P", "Q") *# fa => featured alphabets*

for x in fa:

t = Text(

frames[2],

width = 14,

height = 12,

font = ("Arial", 15),

borderwidth = 2,

relief = "ridge",

padx = 10,

pady = 10)

utb\_simple(t, "Starting from " + x + "...")

texts.append(t)

for i in range(0, len(texts)): utb\_header(texts[i], "\n\n" + getTopNames(fa[i], 10))

*#========================*

*# PACKING WIDGETS FOR EACH FRAME*

*#------------------------*

*# FRAME 1*

promptLabel.grid(column = 0, row = 0, padx = 5)

inp.grid(column = 1, row = 0, padx = 5)

generateButton.grid(column = 2, row = 0, padx = 5)

*#------------------------*

*# FRAME 2*

display.pack(side = "left", expand = True)

*#------------------------*

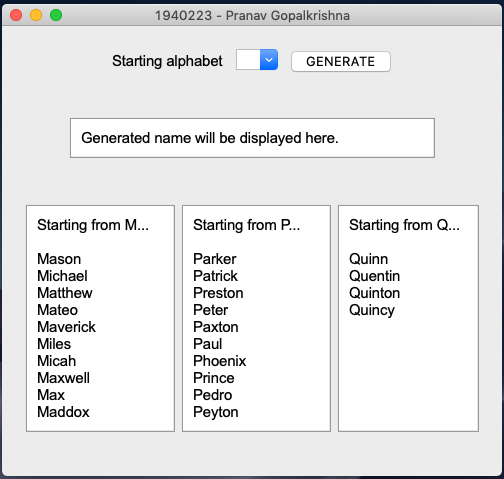
*# FRAME 3*

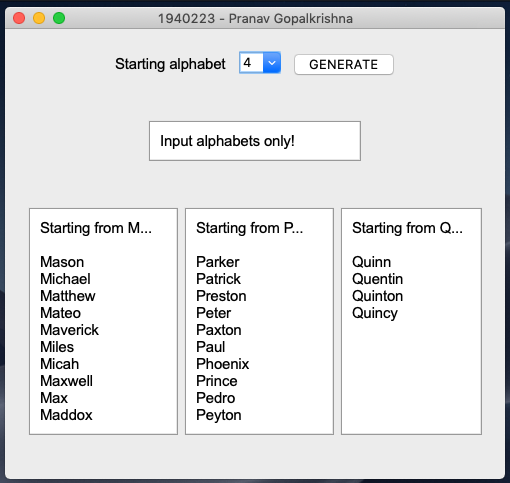
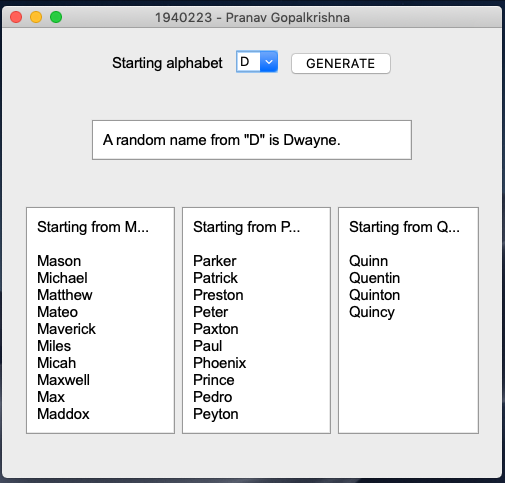
for t in texts: t.pack(side = "left", fill = "both", expand = True)

*#========================*

mainloop()

# Outputs





# Inferences

Within a frame, you must use either grid, place or pack to organize widgets, and you may not mix them, since they are not compatible with each other. However, since two frames are independent widgets, you can use different layout managers for different frames.