

LAB 2402 Sec 351 Graphical User Interfaces (GUI)

GUI Concepts

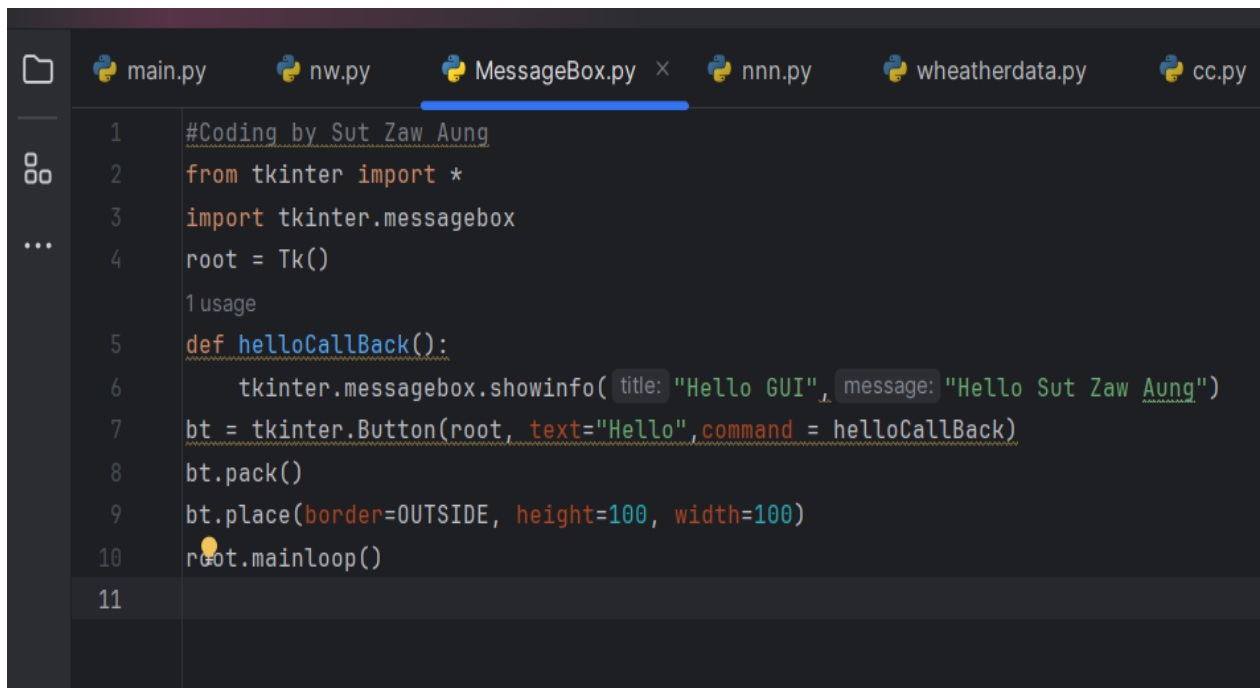
GUI in Python is typically implemented using the Tkinter library, which provides a simple way to create windows, dialogs, buttons, and other graphical elements.

Tkinter follows an event-driven programming paradigm, where the GUI responds to user actions like button clicks or key presses. Developers can define functions, known as event handlers, to execute specific actions when these events occur.

Creating a simple Tkinter GUI involves creating an instance of the Tk class which represents the main window, and then adding various widgets to it. Widgets can be configured with different properties such as text, color, and size, to customize their appearance.

As users interact with the GUI, events trigger the associated event handlers, allowing developers to implement the desired functionality.

Task 1 Write a program to create a window and add button(widget) for showing a message box.

A screenshot of a code editor with a dark theme. The editor has several tabs at the top: 'main.py', 'nw.py', 'MessageBox.py' (which is active and highlighted with a blue underline), 'nnn.py', 'wheatherdata.py', and 'cc.py'. On the left side, there is a sidebar with icons for file explorer, search, and other editor functions. The main area displays Python code for a Tkinter GUI. The code includes comments, imports for Tkinter and messagebox, the creation of a root window, a callback function 'helloCallBack', and the creation and placement of a button. The button is labeled 'Hello' and is configured with a border, height, and width. The code ends with 'root.mainloop()'.

```
1 #Coding by Sut Zaw Aung
2 from tkinter import *
3 import tkinter.messagebox
4 root = Tk()
5
6 #usage
7 def helloCallBack():
8     tkinter.messagebox.showinfo( title= "Hello GUI", message= "Hello Sut Zaw Aung")
9     bt = tkinter.Button(root, text="Hello", command = helloCallBack)
10    bt.pack()
11    bt.place(border=OUTSIDE, height=100, width=100)
12 root.mainloop()
```

Figure 1.1 shown the source code that created the main window, then added the button into it (helloCallback is a function that will be called when the button is clicked)

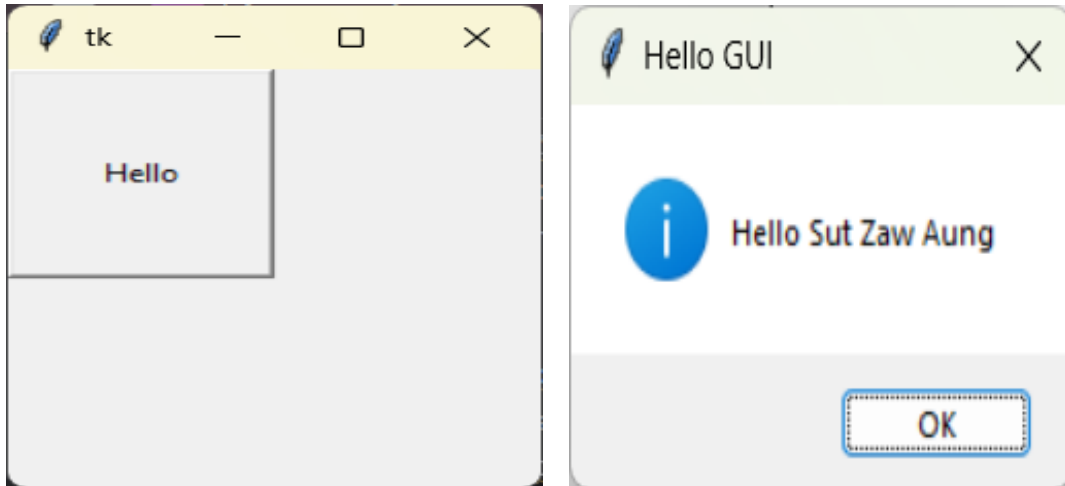


Figure 1.2 Shown the output of the main window and button that users can interact with this GUI, events trigger the associated event handlers.

Task 2 Write a program to create a window and add radio button (widget) into it.

```

PC  PP pythonProject11  Version control
main.py  nw.py  RadioButton.py  MessageBox.py  nnn.py
1  #Coding by Sut Zaw Aung
2  from tkinter import *
3  3 usages
4  def sel():
5      selection = "You selected the option",str(var.get())
6      label.config(text=selection)
7  root = Tk()
8  var = IntVar()
9  rd1 = Radiobutton(root, text="Mr", variable=var, value=1, command=sel)
10 rd1.pack(anchor=W)
11 rd1 = Radiobutton(root, text="Miss", variable=var, value=2, command=sel)
12 rd1.pack(anchor=W)
13 rd1 = Radiobutton(root, text="Mrs", variable=var, value=3, command=sel)
14 rd1.pack(anchor=W)
15 label = Label(root)
16 label.pack()
17 root.mainloop()

```

Figure 2.1 Shown the source code that created the main window and added the radio button into it.

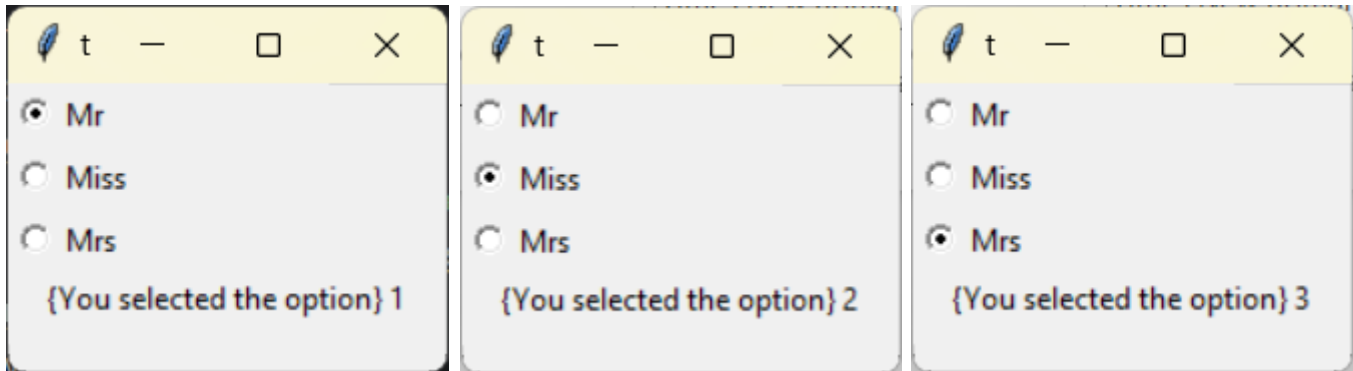


Figure 2.2 when the radio button is selected, the option (label) will be shown.

Task 3 Write a program to create a window and add entry, label, and buttons (widgets) into it.

```

1  #Coding by Sut Zaw Aung
2  from tkinter import *
3  root = Tk()
4  entryVar = StringVar()
5  1 usage
6  def deleteCallBack():
7      ent1.delete( first= 2, last= 4)
8      1 usage
9  def insertCallBack():
10     ent1.insert( index= 3, string= "Hello")
11     1 usage
12 def selectCallBack():
13     ent1.select_range( start= 2, end= 4)
14     1 usage
15 def clearCallBack():
16     ent1.delete( first= 0,END)
17     1 usage
18 def showCallBack():
19     print(entryVar.get())
20     lb1 = Label(root, text="Password: ")
21     lb1.pack(side=LEFT)
22     ent1 = Entry(root, bd= 3, width=10, textvariable=entryVar)
23     ent1.pack(side=LEFT)
24     bt1 = Button(root, text="delete", width=10, command=deleteCallBack)
25     bt1.pack()
26     bt2 = Button(root, text="insert", width=10, command=insertCallBack)
27     bt2.pack()
28     bt3 = Button(root, text="select", width=10, command=selectCallBack)
29     bt3.pack()
30     bt4 = Button(root, text="clear", width=10, command=clearCallBack)
31     bt4.pack()
32     bt5 = Button(root, text="show", width=10, command=showCallBack)
33     bt5.pack()
34     root.mainloop()

```

Figure 3.1 shown the source code that called the five functions by the five button such as, delete, insert, clear, and show buttons.

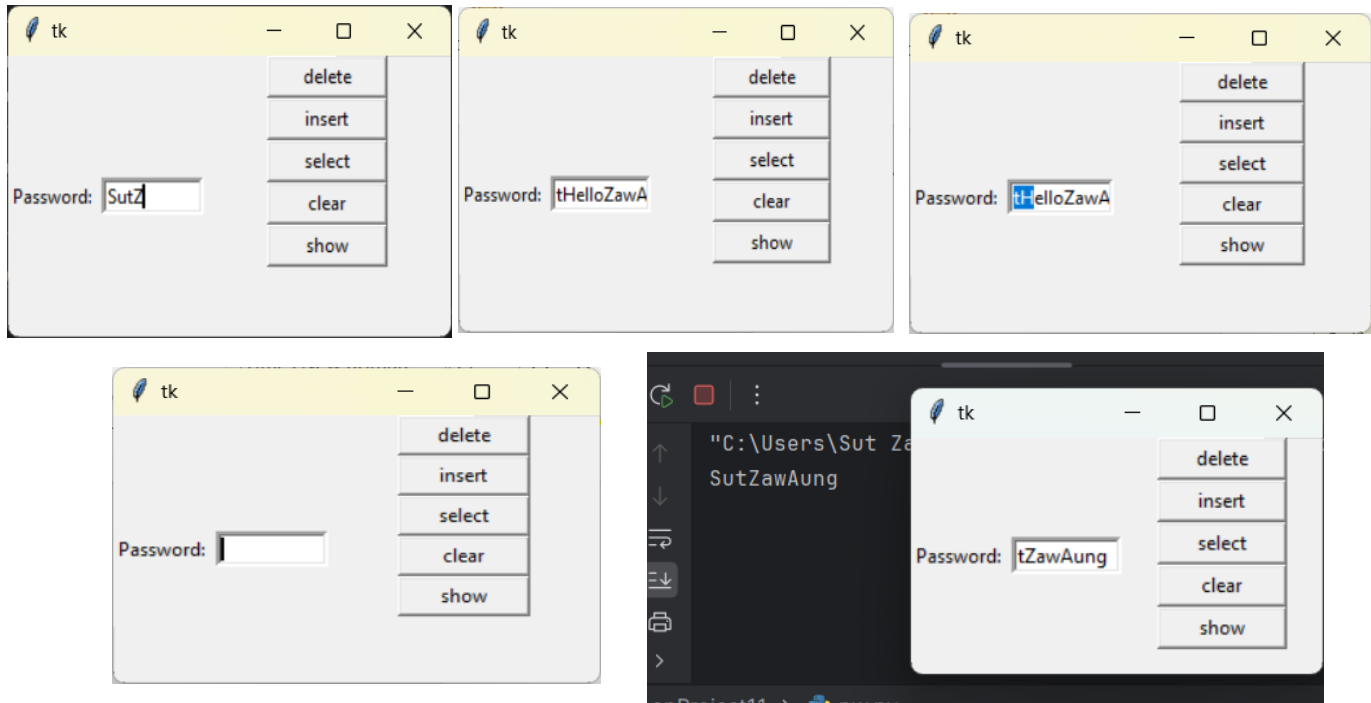


Figure 3.2 shown the output that test the five buttons such as delete, insert, select, clear, and show respectively (all characters will be shown at the termina/console when the “show” button is clicked)

Task 4 Write a program to draw color system.

```

1  #Coding by Sut Zaw Aung
2  from turtle import *
3  import colorsys
4  bgcolor('black')
5  speed(0)
6  pensize(3)
7  hue = 0.0
8  for i in range(300):
9      color = colorsys.hsv_to_rgb(hue, s: 1, v: 1)
10     pencolor(color)
11     hue += 0.005
12     right(i)
13     circle(radius: 50, i)
14     forward(i)
15     left(91)
16 done()

```

Figure 4.1 shown the source code that imported the colorsys module, which provides functions for converting colors between different color systems, set the width of the pen to 3 pixels, and the started a for loop that will iterate 300 times.

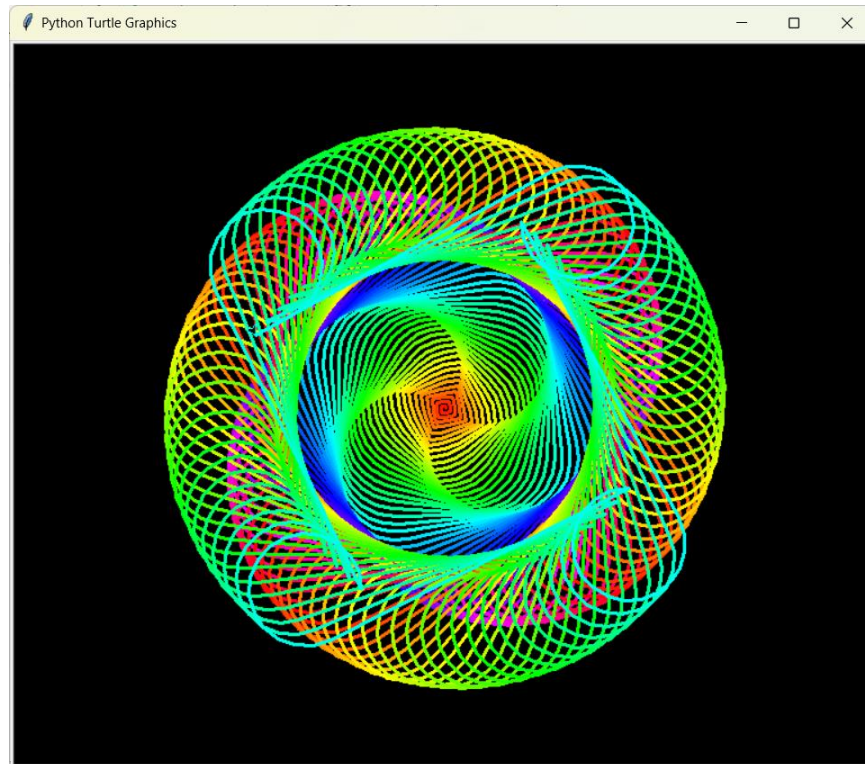


Figure 4.2 shown the output that generated by the functions for converting colors between different color system.