FOSS Lab Experiment : #16

GUI Programming

28 March 2020

Contents

Date : 07/02/2020

1 GUI Calculator

1.1 AIM

To create any application using GUI Programming using any one of Gambas, GTK, QT

1.2 Description

In this experiment we are using GTK tool to make a calculator.
GTK+ is a multi-platform toolkit for creating graphical user interfaces. Offering a complete set of widgets, GTK+ is suitable for projects ranging from small one-off projects to complete application suites.
Calculator also contains some Scientific functions like sin , fact and square root

1.3 GUI Calculator using GTK

Python Code

```
import gi
gi.require_version('Gtk','3.0')
from gi.repository import Gtk
from math import *
import random

class calcWindow(Gtk.Window) :
    def __init__(self):
        Gtk.Window.__init__(self,title="Calculator")
        outerbox = Gtk.Box(spacing=10,orientation =
        ↪  Gtk.Orientation.VERTICAL)
        self.add(outerbox)
        self.entry = Gtk.Entry()
        outerbox.pack_start(self.entry,True,True,0)
        grid = Gtk.Grid()
        outerbox.pack_start(grid,True,True,0)
        button9 = Gtk.Button(label = "9" )
        button8 = Gtk.Button(label= "8" )
        button7 = Gtk.Button(label="7")
        delete = Gtk.Button(label="DEL")
        ac = Gtk.Button(label="AC")
        button4 = Gtk.Button(label="4")
        button5 = Gtk.Button(label="5")
        button6 = Gtk.Button(label="6")
        multiply = Gtk.Button(label= "*")
        divide = Gtk.Button(label= "/" )
        button1 = Gtk.Button(label= "1" )
```

Python Code Continues...

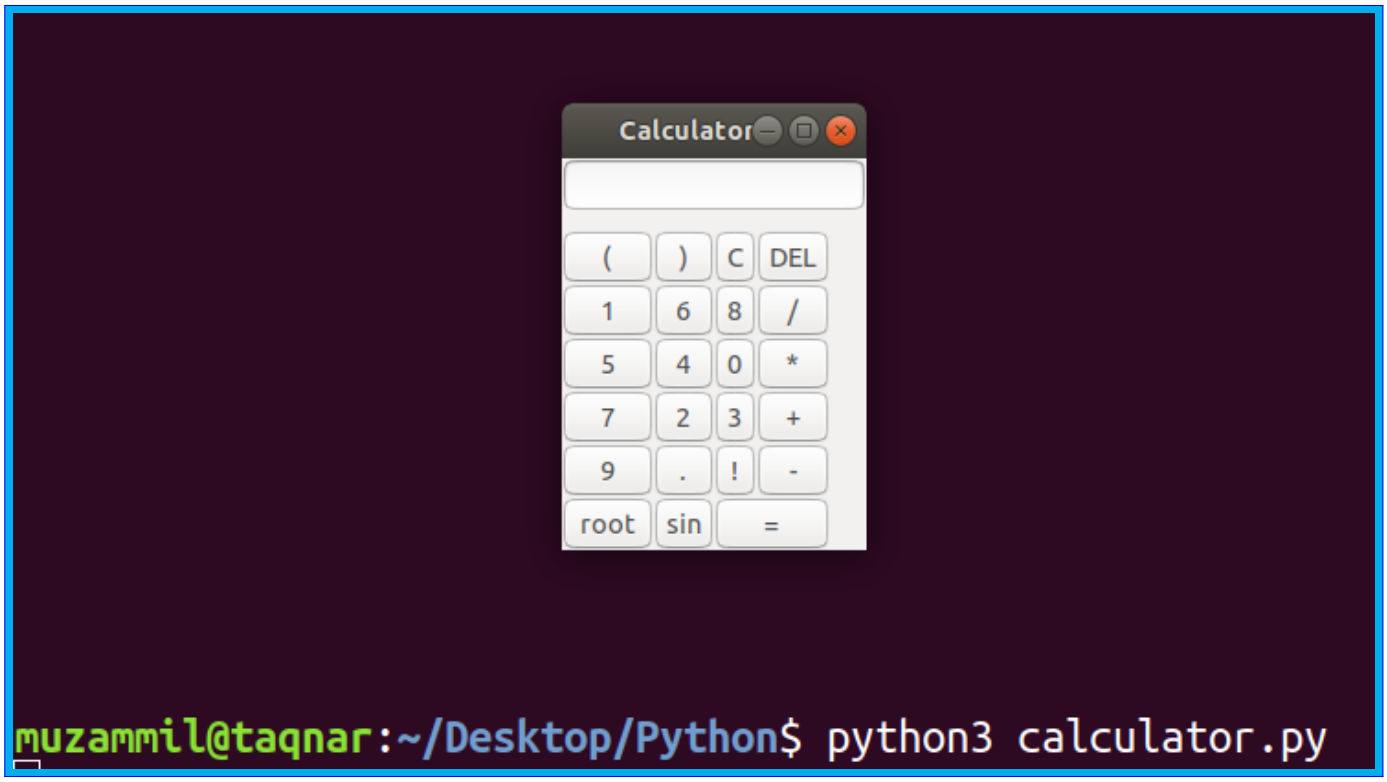
```
button2 = Gtk.Button(label= "2" )
button3 = Gtk.Button(label= "3" )
plus = Gtk.Button(label= "+" )
minus = Gtk.Button(label="-" )
button = Gtk.Button(label= "9" )
ans = Gtk.Button(label= "=" )
dot = Gtk.Button(label="." )
openbr = Gtk.Button(label="(" )
closebr = Gtk.Button(label= ")" )
clear = Gtk.Button(label= "C" )
root = Gtk.Button(label=" root " )
fact = Gtk.Button(label="!" )
sin = Gtk.Button(label="sin" )
button0 = Gtk.Button(label = "0" )
cos = Gtk.Button(label="cos" )
otherbuttons =(divide , plus , minus , multiply )

for buttonname in otherbuttons:
    buttonname.connect('clicked', self.buttonclicked)
digits=[button1 , button2 , button3 , button4 , button5 , button6 ,
 ↪  button7 , button8 ,button9 , button0 , openbr , closebr ]
for i in digits :
    i.connect('clicked',self.buttonclicked)
ans.connect('clicked',self.evaluate)
delete.connect('clicked',self.delsingle)
clear.connect("clicked",self.cleartext)
root.connect('clicked',self.froot)
fact.connect('clicked',self.facto)
sin.connect('clicked',self.sine)
dot.connect('clicked',self.buttonclicked)

grid.attach(openbr, 0,0,1,1)
grid.attach(closebr, 1,0,1,1)
grid.attach(clear, 2,0,1,1)
grid.attach(delete, 3,0,1,1)
grid.attach_next_to(divide,delete,Gtk.PositionType.BOTTOM,1,1)
grid.attach_next_to(multiply,divide,Gtk.PositionType.BOTTOM,1,1)
grid.attach_next_to(plus,multiply,Gtk.PositionType.BOTTOM,1,1)
grid.attach_next_to(minus,plus,Gtk.PositionType.BOTTOM,1,1)
grid.attach_next_to(fact,minus,Gtk.PositionType.LEFT,1,1)
grid.attach_next_to(dot,fact,Gtk.PositionType.LEFT,1,1)
grid.attach_next_to(sin,dot,Gtk.PositionType.BOTTOM,1,1)
grid.attach_next_to(ans,fact,Gtk.PositionType.BOTTOM,2,1)
grid.attach_next_to(root,sin,Gtk.PositionType.LEFT,1,1)

digit=[button1 , button2 , button3 , button4 , button5 , button6 , button7
 ↪  , button8 ,button9 , button0 ]
random.shuffle(digit)
r=1
c=0
```

Python Code Continues...

```
        for button in digit:
            grid.attach(button,c,r,1,1)
            c=(c+1)%3
            if(c==0):
                r=r+1

    def buttonclicked(self, button):
        text = self.entry.get_text()
        text = text+button.props.label
        self.entry.set_text(text)

    def cleartext(self, button):
        self.entry.set_text("")

    def evaluate(self, button) :
        eq = self.entry.get_text()
        try:
            self.entry.set_text(str(eval(eq)))
        except:
            self.entry.set_text("ERROR!!!")
    def delsingle(self, button):
        text = self.entry.get_text()
        text=text [: -1]
        self.entry.set_text(text)
    def froot(self,button):
        text = float(self.entry.get_text())
        text=str(sqrt(text))
        self.entry.set_text(text)
    def facto(self,button):
        text = int(self.entry.get_text())
        text=str(factorial(text))
        self.entry.set_text(text)
    def sine(self,button):
        text = float(self.entry.get_text())
        text=str(sin(text))
        self.entry.set_text(text)


calcWindow = calcWindow ()
calcWindow.connect ('destroy',Gtk.main_quit )
calcWindow.show_all()
Gtk.main()
```

1.4 Function and Output

1.4.1 Output GUI Calculator



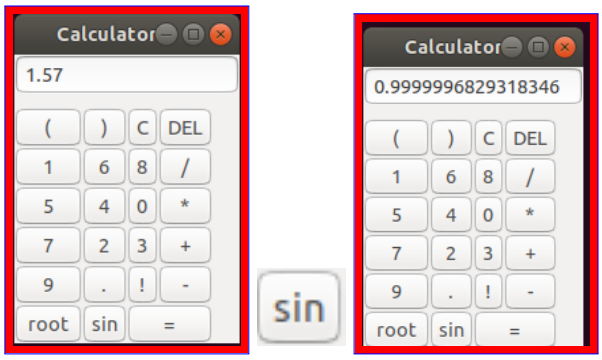
1.4.2 Addition

1.4.3 Subtraction

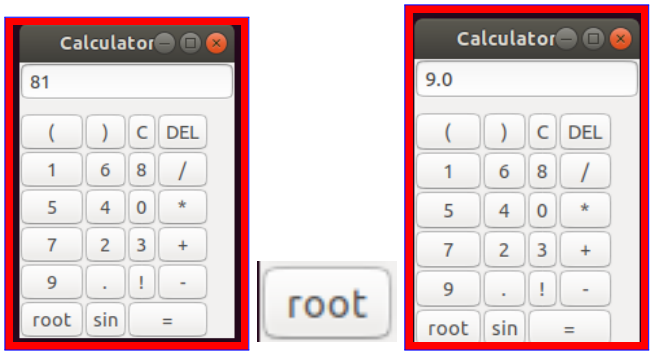
1.4.4 Multiplication

1.4.5 Division

1.4.6 Sin



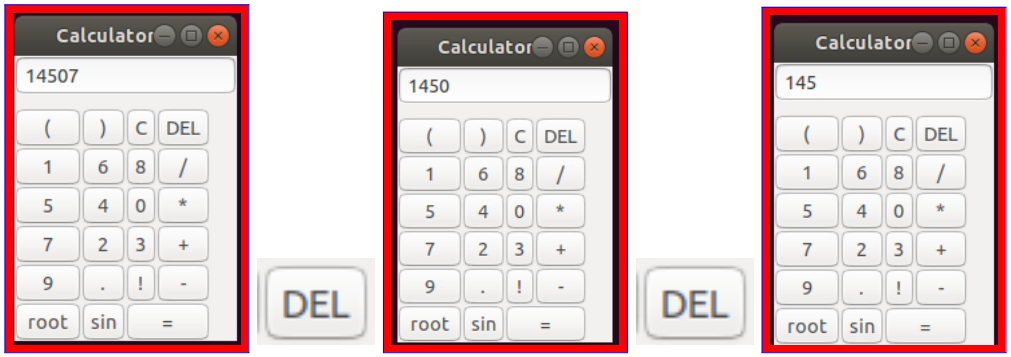
1.4.7 Root



1.4.8 Factorial

1.4.9 Clear

1.4.10 Delete



2 Conclusion

The GUI application - Calculator was made using GTK tool in python and ouput got verified.