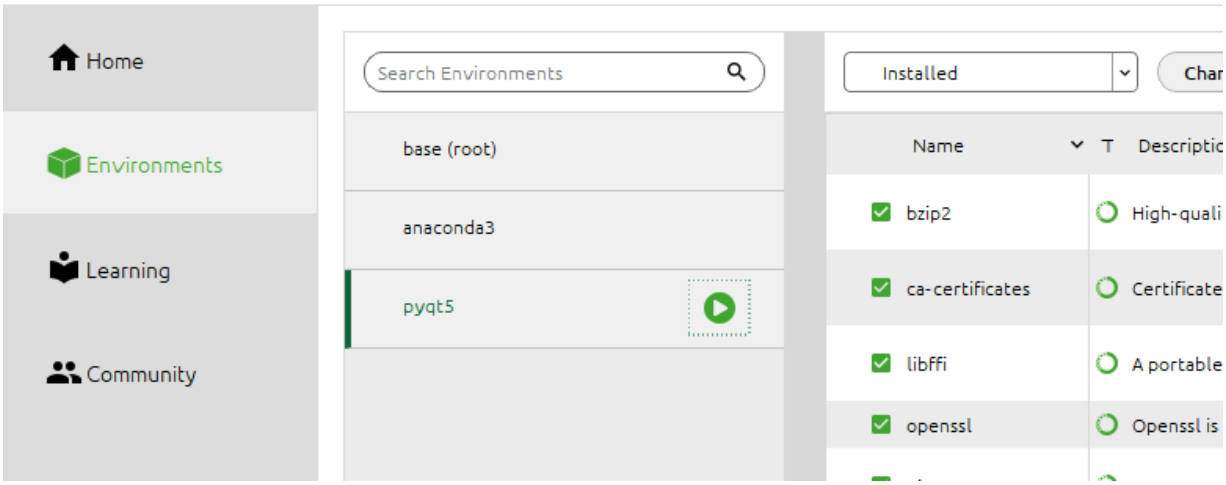
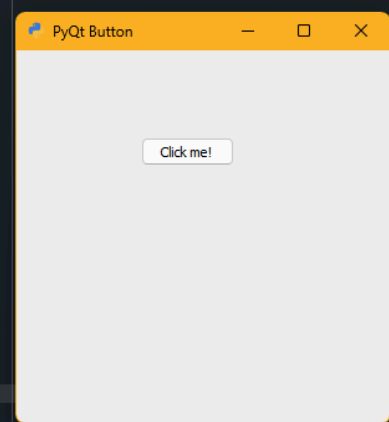
# Laboratory Activity 5 - Introduction to Event Handling in GUI Development

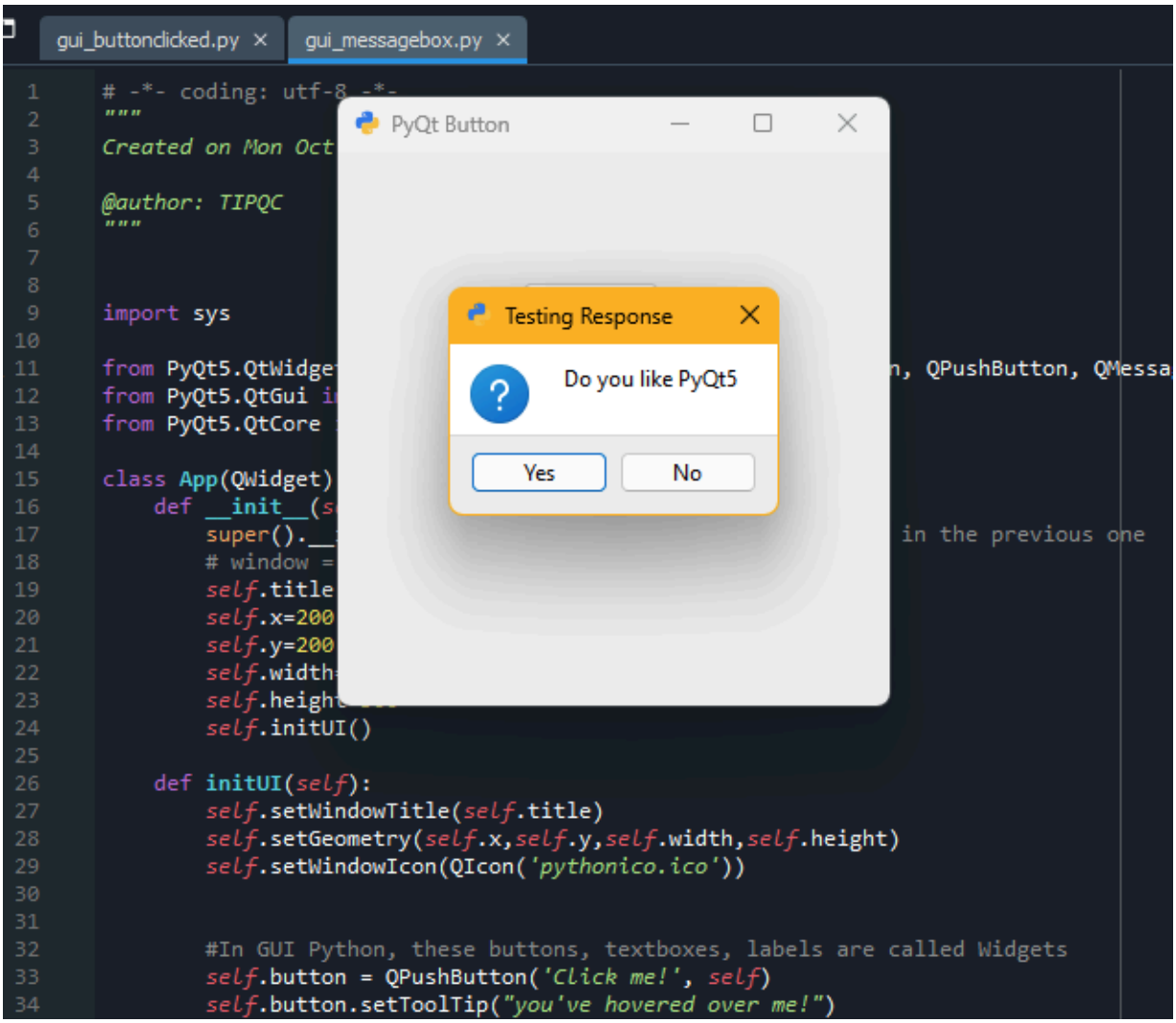
| | |
|---|---|
| Carag, Carl Jervie B. Carag | 21/10/2024 |
| Course/Section : CPE21S4 | Engr. Maria Rizette Sayo |



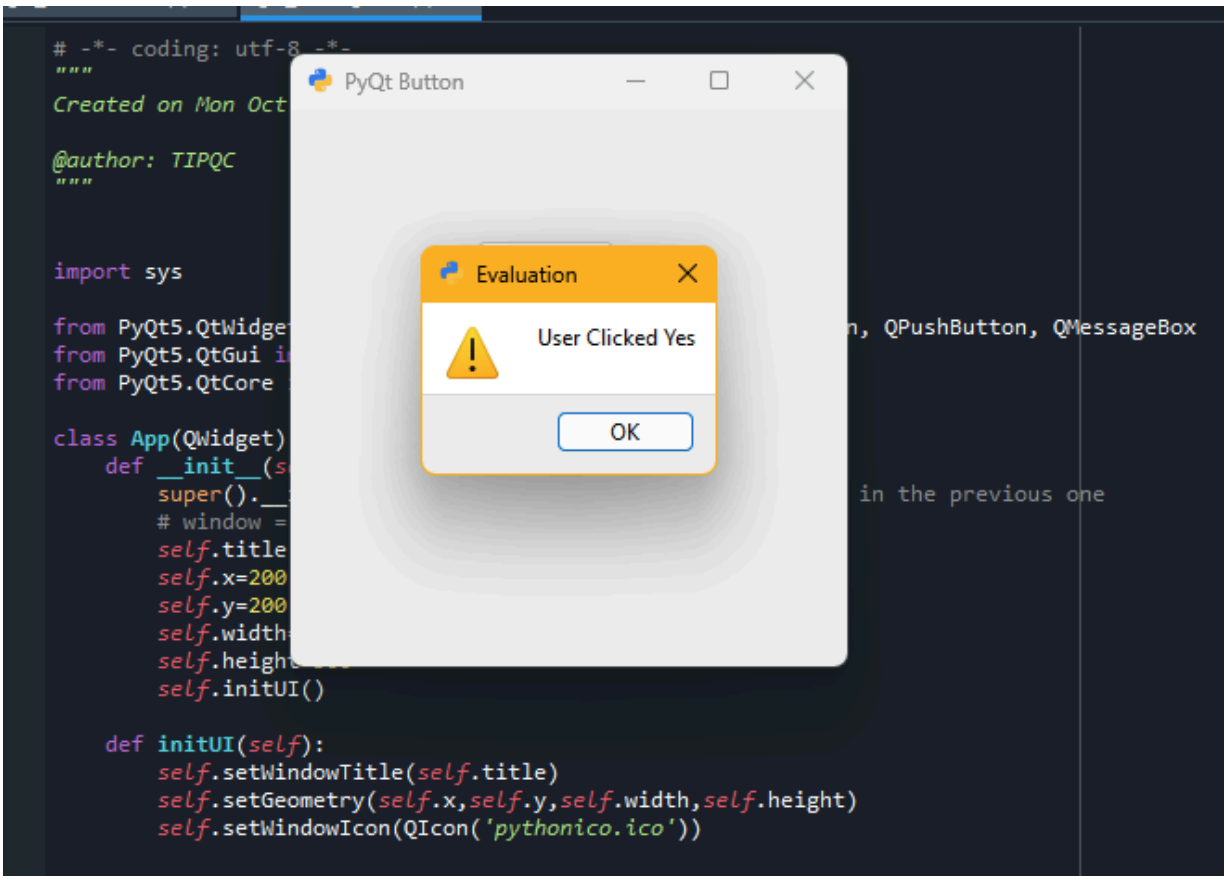
gui_buttonclicked.py × gui_messagebox.py ×

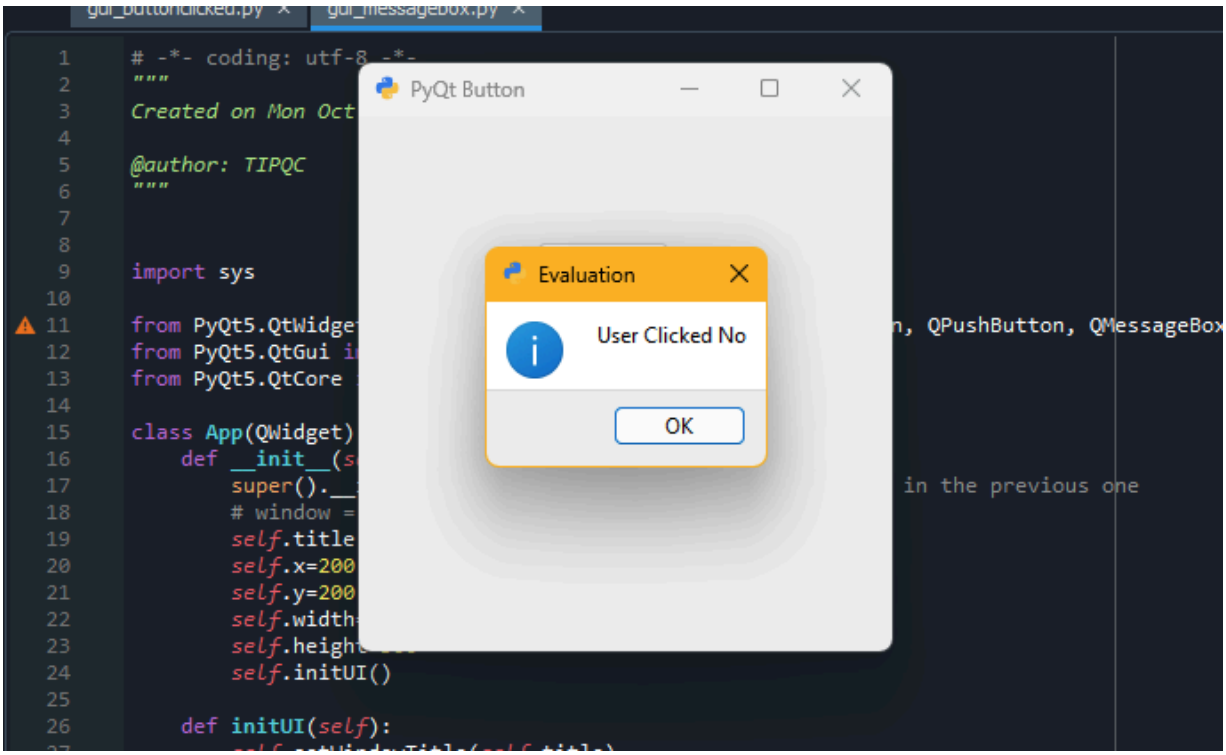
```python
# -*- coding: utf-8 -*-
"""
Created on Mon Oct 21 08:10:15 2024

@author: TIPQC
"""

import sys
from PyQt5.QtCore import pyqtSlot
from PyQt5.QtWidgets import QWidget, QMainWindow, QApplication, QPushButton
from PyQt5.QtGui import QIcon

class App(QWidget):
    def __init__(self):
        super().__init__() #intitializes the main window like in the previous one
        # window = QMainWindow()
        self.title = "PyQt Button"
        self.x=200 # or left
        self.y=200 # or top
        self.width=300
        self.height=300
        self.initUI()

    def initUI(self):
        self.setWindowTitle(self.title)
        self.setGeometry(self.x,self.y,self.width,self.height)
        self.setWindowIcon(QIcon('pythonico.ico'))


        #In GUI Python, these buttons, textboxes, labels are called Widgets
        self.button = QPushButton('Click me!', self)
        self.button.setToolTip("you've hovered over me!")
        self.button.move(100,70) #button.move(x,y)
        self.button.clicked.connect (self.on_click)



        self.show()

    @pyqtSlot()
    def on_click(self):
        print('You Clicked Me!')


if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = App()
    sys.exit(app.exec_())
```

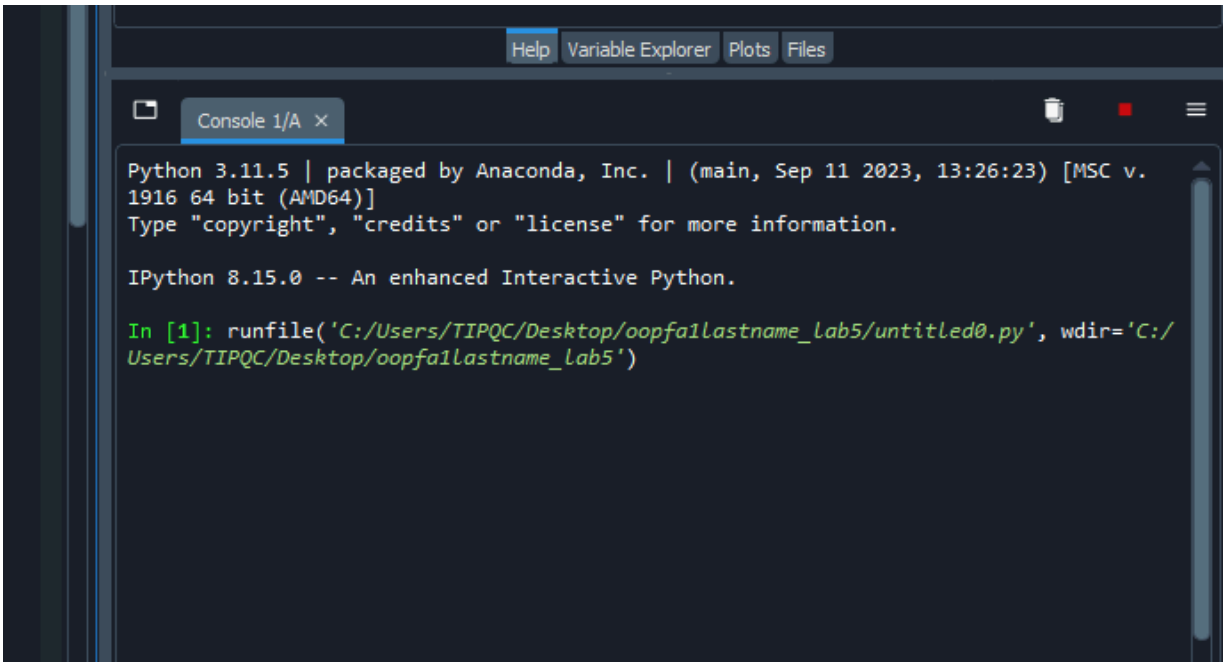





**GUI Button Clicked.py**


PyQt Button
Evaluation
User Clicked Yes
OK


PyQt Button
Evaluation
User Clicked No
OK




**Compiler unsuccessful to Compile the Program (Program will be put in the GIthub)**

1. **What are the other signals available in PyQt5? (give at least 3 and describe each)**

   - These are Qpushbutton signals,.clicked signals, set checked. The QPushbutton enables us to command the computer to perform action and answer questions. The .clicked is responsible for receiving data. the set checked is responsible for storing the data obtained.

2. **Why do you think that event handling in Python is divided into signals and slots?**

- It allows the readability of the codes be more organized and easy to understand. It also allows the different parts of the program to be connected without being mixed with each other.

**3. How can message boxes be used to provide a better User Experience or how can message boxes be used to make a GUI Application more user-friendly?**

- It enables the user to know the statements or the operations that will be performed in the program when they have started to answer the questions asked. It makes them aware of the possible actions taken by the program based on their answer.

**4. What is Error-handling and how was it applied in the task performed?**

- It the way the user may asses and identify the errors that might occur and the errors found in the program, the try, except, and finally are the primary statements the user could do to trap and fixed the error in the program.

**5. What maybe the reasons behind the need to implement error handling?**

- There are a lot of reasons to consider one of the reasons is that it could be a great counter measure when the programs runs incorrectly or there are errors in the program which needs to be found and fixed.

**Conclusion:**

I have identified the different event handling procedures in this activity. Event handling is simply an action, such as clicking a button which then would lead into another event to partake. The importance of event handling was to manage the events and organize them to make them work as they are intended to perform. The way that event handling works in python is initiated by the class that signifies the start of the event and it is also called the “Publisher” and the receiver of the events are called the “Subscriber”, these subscribers may use the operations in the publisher. I have only used the “Publisher and Subscriber” for illustration purposes of how the Event handling in python works. In this activity also the Pyqt5 was used as the environment where our codes will be created. I have used to import the “PyQt5.QtCore import pyqtSlot” to allow the implementation of the code.